A METHODOLOGY FOR THE PREDICTION AND ANALYSIS OF PRECURSORS TO FLIGHT ADVERSE EVENTS

A Dissertation
Presented to
The Academic Faculty

By

Marc-Henri Bleu-Laine

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in Computational Science and Engineering in the
School of Aerospace Engineering

Georgia Institute of Technology

May 2021

A METHODOLOGY FOR THE PREDICTION AND ANALYSIS OF PRECURSORS TO FLIGHT ADVERSE EVENTS

Thesis committee:

Dr. Dimitri Mavris, Advisor
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Tejas Puranik
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Duen Horng Chau
College of Computing
Georgia Institute of Technology

Bryan Matthews
KBR Inc., NASA Ames Research Center

Date approved: April 26, 2021

L’elephant n’echoue pas sur le port de sa trompe

Proverbe Dan

To my beloved family

ACKNOWLEDGMENTS

I would like to start by thanking God for all the opportunities he has provided me since I was born and for giving me the strength to focus and continue moving forward in all the stressful situations that I faced.

I am highly thankful for my family, who supports me, motivates me, and loves me. I thank my parents Gilbert and Simone Bleu-Laine, Delphine Goffri, and my siblings Cynthia, Raymond, and Gilles-Arnaud. Each member of my family has shaped my vision of the world and helped me be a better person. I was taught the value of hard work, humbleness, and the importance of strong family ties. I would have never aimed this far in my life if they had not been with me at every step of my journey since I was a baby.

I would like to express my most profound appreciation to Prof. Dimitri Mavris for taking me as a graduate student at the Aerospace Systems Design Laboratory. The ASDL experience is truly a unique one where first years have class with one of the most knowledgeable professors in the industry, work on many open-ended projects, come across so many opportunities, and create new connections with people that will be friends forever. I am proud to call myself an ASDLer!

I am incredibly thankful to Dr. Tejas Puranik and Mr. Bryan Matthews. They have been meeting with me regularly since the conception of the initial ideas of this thesis. I appreciate the time you invested in me and for believing in the work that I do. I also would like to express my gratitude to Prof. Polo Chau for accepting to be a member of my committee and providing me with his expertise in machine learning.

Finally, I thank Elellta Tesfaye for bringing me a daily dose of sunshine with her fun and positive energy. She helped me stay calm in tough situations with her presence and prayers and gave me valuable advice when I needed it the most. I also thank Eugene Mangortey, now Dr. Mangortey, for all the help he has been providing me since our undergraduate years and for making me discover the world of machine learning. Big thank you to everyone at

the French House and everyone I met at Georgia Tech/ASDL that I can call my friend. It's been a long ride!

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Air transportation is known to be the safest mean of transportation nowadays. The drastic improvements in aviation safety since its gain in popularity are undeniably a factor in the industry's growth over the last several decades. This growth brought social and economic benefits throughout the world and was expected to keep its momentum pre-COVID-19. Stakeholders such as the National Aeronautics and Space Administration (NASA), the Federal Aviation Administration (FAA), the National Transportation Safety Board (NTSB), aircraft manufactures, and airlines have developed systems, techniques, and technologies that are to thank for today's overall safety improvements and the reduction of accidents. The industry's maintained growth is welcomed, but current safety performances have been observed to stagnate instead of declining. With safety initiatives such as the Flight Operational Quality Assurance (FOQA) program and the growing number of aviation data, many of the previous techniques used to understand the causes of accidents are not scalable. These reasons led to the development of novel methods leveraging advanced analytical tools such as machine learning and deep learning. However, current use cases have focused mainly on anomaly detection and system health monitoring, which does not bring enough reaction time to deal with an imminent event. This research proposes the improvement of aviation safety through precursor mining. Precursors are defined as events that are highly correlated to the adverse event that they precede. Therefore, they provide predictive capabilities and can be used to explain pre-defined events. This thesis uses publicly available flight data to 1) develop a novel deep learning method to identify and rank precursors of multiple adverse events, 2) use unsupervised learning algorithms to group flights based on their precursors to identify potential causes for these events at a fleet-level, and finally 3) detect novelty to ensure that the developed precursor models operate within their limits and that new non pre-defined adverse events could be detected.

CHAPTER 1
INTRODUCTION

1.1 Aviation Safety: Overall Statistics

The aviation industry brings many social and economic benefits to human-kind [1]. Over the past decades, the industry size has doubled every 15 years, reaching $4.3 Billion passengers in 2018 [1]. Before COVID-19, the growth was expected to continue with the revenue per passenger kilometer reaching 22 Trillion by 2045 [1]. One of the key catalysts of this growth is the safety improvements that the industry has experienced [1]. The primary role of aviation safety is to prevent deaths due to air travel [2]. For this reason, analysts measure safety by using the number of fatalities per unit of air travel. The units used are usually the number of passenger trips, flight legs, flight miles, hours flown, and more. Incidents or non-fatal accidents, which are likely to trigger a decline in air travel demands [1] are also taken into account when measuring aviation safety. Nowadays, aviation is considered safe, and airplanes especially are considered the safest means of transportation. Indeed, aviation accident and incident rates have seen drastic reductions since flying has become common. As seen on Figure 1.1, the fatality rate for Part 121 decreased from 80.9 fatalities per 10 million persons on board in 1996 to 0.6 in 2019, which was well below the target rate of 5.9 set by the FAA for that year.

1.2 Aviation Safety: What and Where

Aviation accidents come in many forms and at different segments of a flight. The International Air Transport Association (IATA) 2019 report [4] contains insightful statistics painting a better picture of accidents worldwide. Figure 1.2a shows the accident category distribution for a period of 5 years. Within that time, the following accident categories

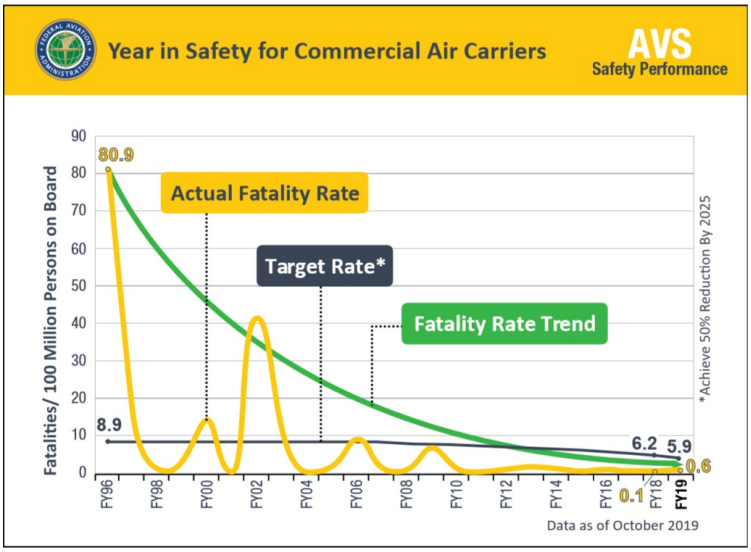


Figure 1.1: Commercial Aviation Fatalities from FY96 to FY19 [3]

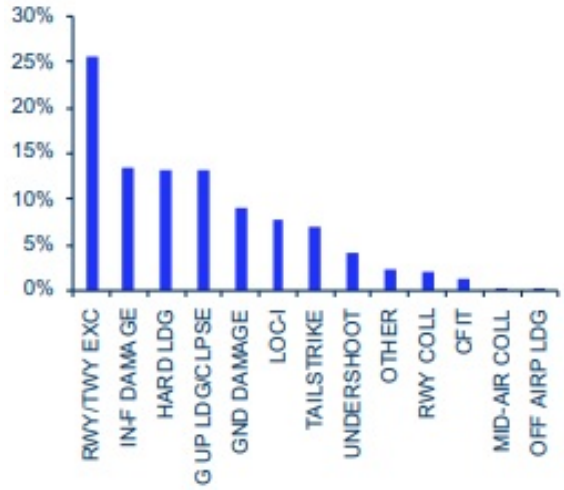
accounted for close to 70% of all accidents:

1. Runway and Taxiway Excursions: Veer off or overrun from the runway surface [5]
2. In-Flight Damage
3. Hard Landing: High vertical speed at touchdown [6]
4. Gear-Up Landing/Gear Collapse

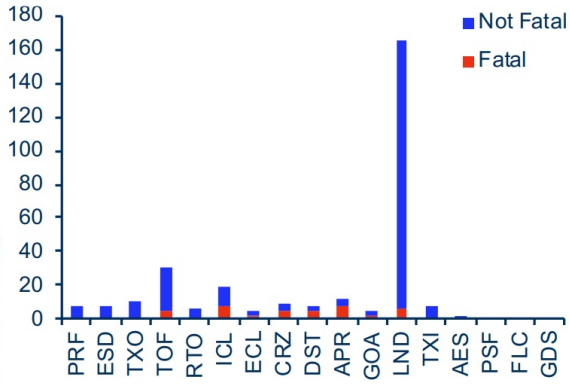
Other accidents include ground damage, loss of control in flight, tail strike undershoot and more. Accidents are also divided into different phases of flights. As seen on Figure 1.2b, crew and passengers were more at risk of an accident during the landing portion of the flights. This higher risk is in agreement with the category distribution presented in Figure 1.2a as runway excursions, hard landings, and gear collapsing are related to this phase of flight.

1.3 Efforts Towards Aviation Safety Improvements

Safety improvements came thanks to the efforts of multiple stakeholders such as the National Aeronautics and Space Administration (NASA), the Federal Aviation Administration



(a) Accident Category Distribution (2015-2019) [4]



(b) Accidents Phase of Flights (2-15-2019) [4]

Figure 1.2: Accidents Categorization

(FAA), the National Transport Safety Board (NTSB), aircraft manufacturers, and airlines [7, 8].

In particular, the NASA and FAA Aviation Safety Report System (ASRS) has enabled the collection of anonymous reports of incidents from pilots, controllers, and others since it was established in 1976 [9]. The analysis of the ASRS reports helps:

1. Identify issues and deficiencies within the National Airspace System (NAS)
2. Support the formulation and planning of policies that are used to improve the NAS
3. Understand the importance of human factors in aviation

In 2016, the ASRS had issued over 2,500 safety alerts to both commercial and private aviation communities, which led 42% of the recipients to improve safety by revising dangerous conditions [10]. NASA's and FAA's efforts ultimately led to better certification standards, better training and operating procedures, better aircraft maintenance, and better decision-making support systems [7].

The NTSB investigations of past accidents that occurred in the United States resulted in reports containing multiple causes of accidents along with any additional contributing

factors [8]. These reports play an essential role in the issuance of recommendations to prevent future accidents [11].

The Flight Operational Quality Assurance (FOQA) is another initiative started by the FAA. It intends to improve safety by providing more significant insights into flight operations [12]. The program is voluntary, and air carriers are responsible for maintaining safe operations conformed to operating standards and regulations. Ultimately, the implementation of the program helps airlines effectively collect operational data and develop methods to analyze data to enhance training for their pilots, review operating procedures, and schedule more efficient maintenance.

Manufacturers also play an essential role in safety improvements. They continuously aim to design and manufacture safer aircraft systems and their subsystems [7]. For example, Boeing design standards have evolved to be more rigorous, incorporate more redundant critical systems, and extensively test the plane's structural strengths[13]. Furthermore, the company continuously monitors aircraft performances so that the manufacturer's engineers can formally analyze safety events. They also implement new technologies, such as predictive wind shear equipment and controlled-flight-into-terrain (CFIT) [13] in aircraft systems, making aviation safer. Human factors are taken into account so that information about human abilities and limitations can be applied to tools, machines, systems, and processes. Finally, they are part of accident investigations when their aircraft is involved [13].

1.4 Motivation

Although safety metrics, in general, are better than decades ago, no significant improvements have been made in recent years. In particular, Figure 1.3 shows the number of accidents (fatal and non-fatal) per 100,000 hours of flight. The rate diminished by half its value of 0.30 in 1999 to 0.15 in 2007. However, since that year, the rate has been roughly constant, which raises concerns. Before COVID-19, more people were expected to fly as the industry was expected to keep growing [1]. Keeping this accident rate while increasing the

number of daily passengers is likely to result in more complex operations and potentially increase the number of people involved in accidents.

Moreover, aircraft systems are changing. The use of composite materials in their design requires different maintenance and inspection procedures instead of aluminum [8]. Newer designs for greater capacities, and longer ranges, additional sensors and new capabilities [14], and aging of regional jets are challenges that need to be overcome to maintain safety records [8].

Although current techniques enabled improvements to aviation safety, they have limitations that seem to restrain further improvement. In particular, current processes are manual [15]. Experts gather to analyze accidents and provide insights and explanations of accidents. The Industry High Level Group (IHLG) expects that by 2026, aircraft systems will generate on average between five and eight terabytes per flight [16]. It goes without saying that a manual approach will not scale well to handle the analysis of such amount of data. In addition to manual processes, lots of the aircraft systems are reactive [17], which results in limited reaction times for pilots in case of an emergency (e.g., vibrating stick announcing stall).

The safety efforts mentioned in section 1.3 all have in common the generation or usage of data. Overall, the aviation industry has been moving towards a proactive approaches by leveraging data mining. Data mining has played a critical part in improving safety. However, most of the applications have been targeting anomaly detection and system health monitoring [18, 19]. More use cases of data mining should be explored to decrease further the accident rate. Hence, recent literature proposes identifying precursors of safety incidents through data mining [17, 20, 21, 22].

1.4.1 Precursor Mining

A precursor is defined in both [15] and [22] as "*any correlated event that occurs before the safety incident with a high likelihood of the safety incident occurring in the future*." The

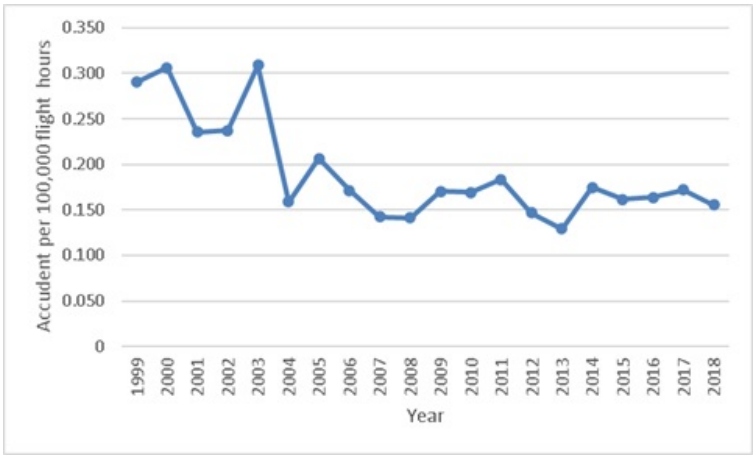


Figure 1.3: Commercial Aviation Accident Rate (1999-2018) [23]

mining of precursors is of great interest because of the advantages it provides [15]:

1. Useful to forecast and prevent safety events

2. Provide insights to why an event occurred

The prevention of the safety event could be performed online. Indeed, since the precursors suggest the near future occurrence of an adverse event, the pilot would be able to perform corrective actions before the event happens. Precursor mining can also be helpful offline as it can be used to investigate the causes of an event after it has already happened by processing historical data. It would then give insights towards how the incident can be avoided in the future or help accelerate investigations [15].

1.5 Summary

Aviation safety has tremendously improved over the past decades thanks to years of continuous efforts by multiple stakeholders composed of governmental agencies and the aerospace industry-leading companies. Technological advancement allowed for safer aircraft; progress has been made in collecting, investigating, and analyzing accidents and their data using more modern tools. While all these signs of progress are great, the aviation industry is still growing, which means that more people flying will need to be kept safe, and current stag-

nating accident rates must be lowered. Modern aircraft technology and a growing amount of sensors might change what is known about aircraft accidents, how they are investigated, and challenge current methods used to analyze flight data. The aviation industry has been moving from reactive to more proactive and predictive systems and methodologies, but many use cases have focused only on anomaly detection and health monitoring. Precursors mining has recently triggered more research interest due to its benefits and possible outcomes, significantly improving aviation safety. These observations motivates the overall research objective of this thesis:

Research Objective:

Develop a data-driven methodology that will help expand proactive approaches in aviation, and improve flight safety by enhancing modern algorithms used to identify precursors of adverse events, leveraging the identified precursors to retrieve the potential causes of the events, and ensuring that the developed algorithm will be used in a real-world setting within its limits.

CHAPTER 2
BACKGROUND AND LITERATURE REVIEW

The aviation industry generates a large amount of data. Today's aircraft have multiple sensors collecting different information such as airspeed, altitude, aerodynamic surfaces' positions, aircraft's attitude to landing gear positions, auto-pilot activation, and more [15]. Reports collected in programs such as the ASRS contain textual information about multiple flights. Consequently, data mining comes as an inherent method that can be leveraged to determine the causes of an important event, their probabilities of occurring, and even detect behaviors that are abnormal [24]. As previously mentioned, multiple research projects have been exploring advanced analytics [25, 26, 27, 19] in aviation for anomaly detection and health monitoring. These applications are possible thanks to safety programs that promoted the collection and creation of data sets. In particular, key enablers that can be leveraged to improve safety are the FOQA program, which provides a rich data set and advanced data analytics tools such as machine and deep learning.

2.1 Flight Operational Quality Assurance (FOQA) Program

The Flight Operational Quality Assurance (FOQA) is a voluntary program endorsed by the FAA[12], in which air carriers can enroll. It is one of the most widespread programs for quantitative safety assessment [28]. When implemented, data generated during flight operations can be automatically recorded by the on-board recorders such as Flight Data Recorders (FDR) and analyzed to get better insights into the operations. Many benefits usually come from implementing the FOQA program [12]:

1. Improved operations

2. Improved training

3. Improved procedures and practices

4. Identification of deficiencies and trends

5. Improvement equipment and infrastructure

In addition to these benefits, the program's implementation also allows for better design and operations of Air Traffic Control (ATC) systems and assists manufacturers in aircraft design [12]. The usual FOQA data is extremely rich as it contains a high number of recorded parameters that are heterogeneous (characterized by different data types such as text, discrete, and continuous variables) [20]. The continuous variables consist of time series describing the aircraft state (airspeed, position, and more.), its attitude (roll, bank, and others.), control surface deflections, and more [29, 20]. Discrete variable examples are the activation of the auto-pilot, gears position, and others. A FOQA program rich data set is leveraged by performing two main types of analysis: exceedance and statistical analyses.

2.1.1 Exceedance Analysis

As the name suggests, this type of analysis involves setting a limit to a particular parameter and observe if the parameter falls outside of the normal operating conditions, i.e., whether it is greater or lower than the specified limit. This methodology implies that a pre-defined event is undesirable [28], and that aircraft in an abnormal condition experience an adverse event. The operators set levels of exceedance for particular events, and higher levels correspond to higher operational risk [12]. The Advisory Circular (AC) [12] defines multiple types of events, and potential thresholds/limits for specific parameters are left out blank for the operators to fill when developing their operations manuals. For instance, AC-120 defines an Approach Speed High event by observing if the aircraft exceeds its computed final approach speed. The AC also provides the rule to define the event:

1. The event occurs for Height Above Takeoff/Touchdown (HAT) greater than 1,000 feet but lower than 3,000 feet, the Computed Air Speed (CAS) is greater than the

reference speed minus some constant speed x to be defined by the operator

2. The event occurs for HAT lower than 1,000 feet if the CAS is greater than the reference speed added to some constant speed x

2.1.2 Statistical Analysis

The statistical analysis looks at trends across all flights operated by a carrier and its total performances. The analysis allows for the creation of data distributions from which risk can be determined based on the means of these distributions and based on how far from the means a flight is [12], which is different from the exceedance analysis that determines risk based on exceedance levels. This data enables carriers to create flight profiles, along with maintenance and operational procedures, allowing them to evaluate their overall operational performances.

2.2 Advanced Data Analytics

Advanced data analytics is a broad area that includes multiple approaches, from Artificial Intelligence (AI) augmented analytics to real-time and predictive analytics, which provides quasi-instant insights [30]. Leveraging the power of advanced analytics to understand the present and the future has never been easier. Nowadays, large volumes of data are available within different fields and industries, algorithms are more sophisticated, faster, and can handle more extensive and more heterogeneous data sets [31, 30]. Furthermore, computational power has increased, and storage capacities have improved [31]. All these improvements led to the advancement of a new big data era [32], where data is a crucial asset and is even defined as the new oil [33]. The big data revolution is characterized by the three V's: volume, velocity, and variety. Big data are large volumes of data requiring lots of storage, and while this data varies in its format (numerical, images, text, and more), it also streams at faster speeds than ever before [34]. Machine learning and deep learning are critical tools used when implementing advanced data analytics on big data.

2.2.1 Machine Learning

Machine learning is particularly useful when dealing with data, as it is composed of a set of methods that can be used to find patterns in it automatically[32]. Another insightful definition for machine learning is the study of algorithms that improve their performances at some tasks with experience [35]. Machine learning has proven to be an effective tool for many applications such as improving decision making for many industries, fraud detection, cancer diagnosis, recommendation systems, voice assistant, and more [36]. Machine learning can be divided into two main learning subtypes: supervised and unsupervised learning [32].

Supervised Learning

The supervised learning type enables a machine learning model to learn the mapping from an input x to an output y, using a training set composed of input-output pairs [32]. The inputs are commonly referred to as features or predictors, and the outputs as labels or targets. There are two possible learning tasks when working with labeled data:

- **Classification:** Class labels are provided to the machine learning model [18, 32]. If the targets include two classes, then the task is a binary classification task, while more than two mutually exclusive classes turn the task into a multi-class classification. In the case of mutually inclusive classes, the task is referred to as multi-label. Examples of classification problems include fraud detection, handwritten digit recognition, speech tagging, and more.

- **Regression:** Similarly to classification, labels are provided for the regression task. However, the labels are continuous variables instead of just classes. Examples of regression tasks include predicting stock market price, predicting the number of COVID-19 cases, predicting the age of viewers given a YouTube video, etc.

For each of these tasks, the machine learning model's learning process usually includes

using data instances from a training set to make predictions and comparing them to the real values, i.e., the targets/labels of these data instances. A cost (also called loss) function measures how far the model predictions are from the actual target values by computing an error. Finally, the model uses the error to update its parameters, usually by leveraging the gradient descent algorithm to minimize the loss and make better predictions the next time it sees the same data instances. The process is repeated for a defined number of iterations. The Model performances are usually evaluated using a given metric depending on the task. Common metrics and machine learning models for the classification and regression tasks are presented in table Table 2.1

Table 2.1: Common Metrics and Supervised Machine Learning Models [18, 37, 38, 39]

Task	**Metrics**	**Machine Learning Algorithm**
Classification	Confusion Matrix Accuracy Precision Recall F1 Score AUC	Logistics Regression Decision Trees Support Vector Machine K-Nearest Neighbor Naive Bayes
Regression	Mean Squared Error Mean Absolute Error	Linear Regression Regression Tree Support Vector Regressor

Unsupervised Learning

The unsupervised learning task aims to discover knowledge from a data set [32]. Unlike supervised learning, the data is not labeled, and the machine learning algorithms rely solemnly on the structure of the input x to create groups of similar data points (clustering), determine the distribution of the data within the input space (density estimation), or to reduce high dimensional data to lower 2 or 3-dimensional data sets for visualization [40]. These algorithms are in practice applicable to more use cases as no human expert needs to label the data set, resulting in more available data. Gavrilovski et al. [18] propose the

following taxonomy for unsupervised tasks:

- **Clustering:** Use structured unlabeled data to objectively organize it into homogeneous groups. The algorithms maximize the within-group similarity while minimizing the between-group one. Distance metrics, such as the Euclidean distance, are usually used to compute the similarities. A typical example of clustering is customer segmentation for better marketing

- **Association:** The association rule mining task finds "if-then" rules that express significant associations among the features. The associations represent dependencies and relationships within the data, which can provide useful insights for decision-making. Applications of association rule mining are market-basket analysis, cross-marketing catalog design, recommendation systems, and more [41]

Common unsupervised learning algorithms are presented in table Table 2.2.

Table 2.2: Common Unsupervised Machine Learning Models [18, 38, 42, 43, 44]

Task	**Machine Learning Algorithm**
Clustering	K Means DBSCAN Hierarchical clustering Gaussian Mixture Models Hidden Markov Models
Association	Apriori ECLAT Frequent Pattern-growth

2.2.2 Deep Learning

Deep learning is a subset of machine learning, taking advantage of large Artificial Neural Networks (ANN). Like machine learning, supervised and unsupervised learning methods are also applicable in deep learning, though different algorithms are used. Indeed, deep

learning can also be used for classification, regression, and clustering tasks. Deep learning implementations have multiple advantages as they tend to have better performances, with enough data, and do not theoretically require domain experts since the learning process allows the algorithm to learn information independently. Additionally, one can think of deep learning architecture components as ”lego blocks” that can be added together as wanted, hence allowing for modularity and plug-and-play architectures. Deep learning methods require the presence of three concepts [45]:

- Hierarchical Compositionality
- End-to-End Learning
- Distributed Representations

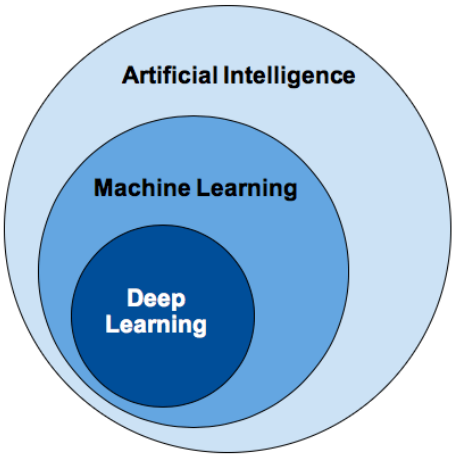


Figure 2.1: Artificial Intelligence Subsets[45]

Hierarchical Compositionality

Deep learning models are complicated functions resulting from the compositions of multiple more straightforward functions. They usually involve several layers of function compositions and non-linear transformations enabling multiple layers of representations [45]. Figure 2.2 shows a deep learning architecture composed of three layers. The model takes

an image as an input and applies a function on the first layer from which the input image's low-level features are extracted. The first layer's output is the input to the second layer that applies another function, hence the functional composition. The cascade of layers enables the model to learn important features, enabling the recognition of the object in the original input image.

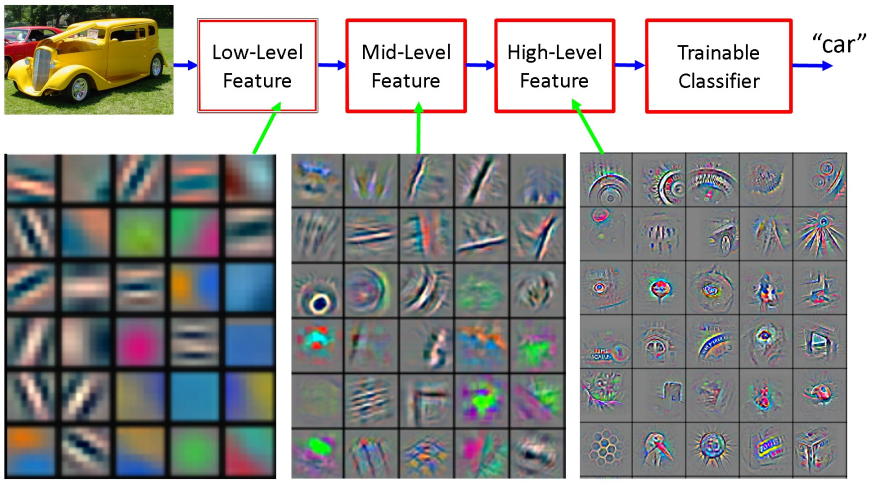


Figure 2.2: Hierarchical Compositionality of Deep Learning[45, 46]

End-to-End Learning

Traditional machine learning techniques require handcrafted features through manual work. After performing feature engineering to the input data, the processed data is passed to the learning algorithm as seen on Figure 2.3. Multiple processing stages may be required, and the engineered features are limited by the knowledge/expertise of the human preparing the data. End-to-End learning replaces the different pre-processing stages into one pipeline [45, 47], in which the data is sent to the algorithm, which learns the important features itself.

Distributed Representations

Deep learning architectures rely on neural networks as a group; no single neuron in the network encodes all the information [45]. The distributed representation entails that many-

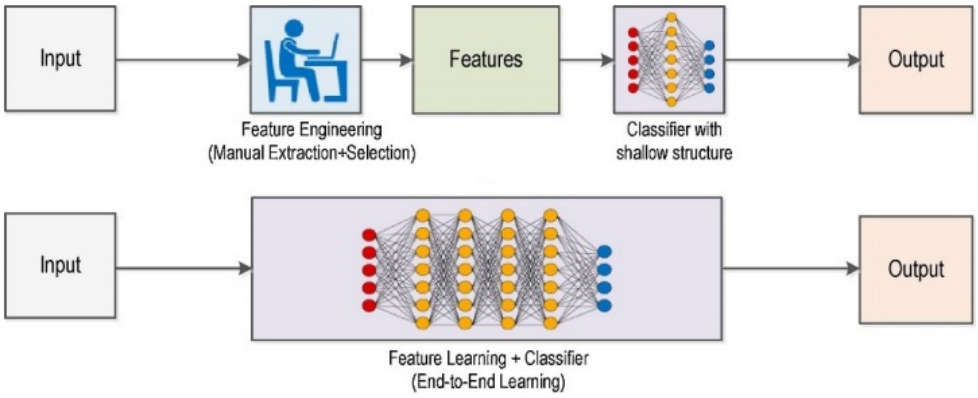


Figure 2.3: End-To-End Learning vs.. Traditional Machine Learning [48]

to-many relationships exist between neurons in the network, so that together many neurons learn to represent a given concept, enabling the network as a whole to represents many concepts [49].

2.3 Review of Prior Applications of Advanced Data Analytics to Aviation for Safety Improvement

The aerospace industry gathers a large volume of data from data recorded at the individual aircraft level to data recorded over the whole National Airspace System (NAS). This amount of data and the advancement of analytics led to exciting applications of machine learning and deep learning to improve aviation safety. It can be seen from surveys [18, 19] of past research using advanced data mining techniques that the use cases have mainly involved applying these techniques to anomaly detection, system health monitoring, and predictive maintenance while few recent projects have moved towards precursors identification.

2.3.1 Anomaly Detection

Anomaly or outlier detection is a task in which new or unknown patterns are identified [50]. These patterns are usually different from normal behaviors, and discovering them can provide insights into operational conditions. Detecting the anomalies is a difficult task to

perform. Data sets are usually imbalanced, meaning that since anomalies are rare events, only a few instances of them are labeled [19]. Furthermore, it can be difficult to distinguish between normal and abnormal as boundaries between them are sometimes not clear and change over time. Anomalies can be divided into three types [26, 19]:

- **Point Anomalies:** An instance of the data (or data point) is different from all other data points
- **Contextual Anomalies:** A data point that is abnormal only in the current given context
- **Collective Anomalies:** A group of data points is an anomaly as a whole, though the individual instances might not be considered anomalous if they were on their own. This type of anomaly can only exist in sequential, spatial, and graph data

One of the most successful implementations of anomaly detection algorithms in aviation is the Multiple Kernel Anomaly Detection (MKAD) algorithms developed by NASA's Intelligent Systems Division [19, 51]. The algorithm can be used to detect anomalies in FOQA data. Examples of anomalies studied are high airspeed events, go-around operations, abnormal approaches, and gust impacted flights. The algorithm handles sequential discrete and continuous data types and leverages a One-Class Support Vector Machine (OC-SVM) model to identify anomalies in operational data in a supervised manner. Providing kernel functions to the SVM model allows for non-linear decision boundaries for the classification task.

Orca [52] is another popular anomaly detection algorithm. The algorithm implements a K-Nearest Neighbors (KNN) machine learning Model to detect outliers [51, 52]. For this algorithm, the distances of pairs of points are calculated. Indeed, Orca uses the average distance of a point to its KNN to determine how close the point is to the others in the neighborhoods. In order to improve on the algorithm time, the authors developed a pruning method. Orca also handles both continuous and discrete data types

Sheridan et al. [25] used a hierarchical-based clustering method to categorize airports and then implemented the Density-Based-Spatial Clustering (DBSCAN) algorithm, another distance-based method, on FOQA data to identify anomalies during the approach phase. DBSCAN uses the number of points within the neighbors of a seed point to create clusters. Two important parameters for this algorithm are the minimum number of points around the point of interest and the maximum distance ϵ from one point to another. The first parameter is used to determine if a point is a seed point, and the second is used to determine the neighborhood of a point. Clusters are formed depending on these two parameter interactions, and points in low-density regions are flagged as outliers. In addition to finding anomalies, the authors quantified the anomalous behavior for each flight using an anomaly score and discovered that abnormal behavior was strongly related to the approach phase length.

Cluster-based Anomaly Detection (ClusterAD) is another anomaly detection method levering the DBSCAN algorithm [18, 53]. Aircraft parameters for a specific phase of flight are analyzed across multiple flights to identify anomalies. The algorithm was implemented for take-off and approach phases, and detected reduced power and power change anomalies for the take-off phase, and identified low speed and unusual flap setting conditions during the approach.

SequenceMiner compares a given set of discrete sequential data using the normalized longest common sub-sequence (LCS) [51, 18]. LCS measures the similarity between two sequences. Therefore, the length is a similarity score allowing the clustering of similar sequences together, and sequences far from the clusters' medoids are considered outliers.

2.3.2 System Health Monitoring and Predictive Maintenance

In aviation, aircraft maintenance is required to keep the crew and passengers safe. Maintenance helps keep an aircraft airworthy by restoring and maintaining the aircraft's systems, components, and structures [54]. Three main reasons make maintenance mandatory:

- **Operational:** Deals with maintaining the aircraft in proper conditions to continue service
- **Value Retention:** Minimizes the degradation of the aircraft such that its value could be maintained
- **Regulatory Requirements:** Regulatory authorities have established standards for repair, periodic overhauls, and alteration. The aircraft operator is required to conform to those requirements.

Predictive maintenance is used when the scheduling of maintenance activities depends on predicting the failure time of an aircraft and can be divided into preventive and conditional maintenance [55]. Thanks to the onboard aircraft sensors, the data they generated in combination with maintenance data can be used for fault diagnostics, particularly for the computation of the Remaining Useful Life (RUL). Concepts for prognostics and health management (PHM) are commonly applied to predictive and condition-based maintenance [19]. The two significant advantages of implementing predictive capabilities for maintenance are efficient maintenance schedules and catastrophic or disastrous failure prevention.

Gugulothu et al [27] developed a novel approach for RUL and health index estimations on a publicly available engine data set and a pump data set. The approach can handle noisy sensors and missing data and involves using and training in an unsupervised manner a Recurrent Neural Network (RNN) encoder-decoder to generate embedding of time series data. These embeddings are created for both standard and degraded systems, revealing patterns that can be used to observe similarities between multiple machines' operational behaviors. The machine health can be quantified by comparing a new recent embedding to historical embeddings of normal behaviors.

Deep Belief Network (DBN) is another algorithm that has been applied to diagnose and classify aircraft engine health states [56]. The classifier is composed of stacked Restricted Boltzmann Machines, which enables a layer to layer learning process. The data set was

collected from an aircraft engine dynamic simulation, and it includes the operating cycle index, operational settings, and 21 sensor readings. The authors implemented a three stages approach that encompasses the pre-prossessing of multiple sensor data, then training and validating the DBN to predict pre-defined health states. The DBN algorithm was compared to other algorithms and generally performed better at detecting the health states according to the classification accuracies.

Altay et al. [55] used ANNs and Genetic Algorithms (GA) to predict an aircraft's failure based on its type and age, previous failures, and their time of occurrence. A regression task is performed with the target being the maintenance times requirement for aircraft, and the comparison of the two algorithms used for this tasks are also compared, and the ANN provided slightly better results with a correlation between the target and predicted value of 0.8977.

Nicchiotti et al. [57] use BIT messages, flight deck effects, and logs of maintenance activities to forecast failure events within prediction windows of two to ten flights in a supervised manner through a classification task. Since two flights are the minimum notice period and ten flights are the maximum, operational disruptions are reduced. The authors used an SVM first to detect flight legs with prognostics alert. Subsequent steps used the Eigenface technique, a computer vision technique, to generate signatures of multiple types of maintenance actions. The technique uses the Principal Component Analysis (PCA) to detect variances within the data and can encode and compare the different signatures [58]. Finally, the component to be replaced is identified using a template matching algorithm. The final performances of the algorithm showed high precision scores but low recall.

2.3.3 Precursor Mining

As previously mentioned, on the one hand, anomaly detection, system health monitoring and predictive maintenance have been in the spotlight for the application of tools such as machine learning and deep learning to improve aviation safety. On the other hand,

precursor mining applications are relatively recent, and limited literature is available on the topic.

Janakiraman et al. [15] define a precursor as an event that is highly correlated with adverse events that occur before the adverse event itself. From this definition, the advantages of precursor mining are straightforward to see, as highlighted in subsection 1.4.1. The precursor mining technique developed by the authors leverages a Deep Temporal Multiple-Instance Learning (DT-MIL) framework, which resulted in the Automatic Discovery of Precursors in Time Series Data (ADOPT) algorithm. The algorithm combines Multiple-Instance Learning (MIL) and RNNs such as a Gated Recurrent Unit (GRU). MIL framework is beneficial when dealing with a lack of validated labels within a data set. This framework defines two concepts of importance, namely bags and instances. Bags are sets composed of multiple instances and can be thought of as a given flight in this context. Instances in the bag are the individual time-steps that composed the flight. In the MIL framework, the label is only available at the bag level (e.g., an adverse event for the flight) such that the bag can be labeled positive or negative. Instances can also be labeled in such way. A common assumption is to assume that bags are positive if at least one instance in the bag is. ADOPT leverages the temporal pattern recognition abilities of its GRU to retrieve time instances (time steps) that are positives within a positive bag (i.e., a flight that experienced an adverse event). The algorithm, therefore, assigns probabilities (referred to as precursor score) to each time step. The probabilities relate to the possibility of an adverse event occurring. Moreover, the algorithm uses a pre-defined threshold to classify the instances, and therefore the bag, as positive. A sensitivity analysis is later performed, and each input feature to the algorithm is perturbed one at a time. The precursors are identified by determining which feature's perturbation had the most impact on the precursor score.

Ackley et al. [20] developed a methodology that leverages a Sequential Backward Selection (SBS) with a Random Forest classifier to predict unstable approach adverse events and determine their precursors. The SBS is typically used for feature selection as it re-

moves features sequentially according to selected criteria. When applying this algorithm, the authors start with f features and train all possible models with $f - 1$ features, then select the best model according to its performance on a testing set for the next iteration. The process is repeated until a pre-specified number of features k is reached. To train the Random Forest using the FOQA data, the authors created a point-specific feature vector for each flight, representing a snapshot from a time point or a given altitude. The feature vector elements are composed of the flight parameters at that snapshot. Feature vectors for multiple flights were created and concatenated with each other to create the data set that can be used to train the Random Forest in a supervised manner. The feature vector formulation allowed the authors to train models at different altitudes during the approach phase. Indeed, at each altitude, the implemented algorithm trains the Random Forest multiple times using the SBS. The final results are models able to predict the adverse event at different altitudes. The Random Forest interpretability is leveraged to discover the most critical parameters at each of the altitudes, which can be thought of as the progressional changes of precursors of the event.

More recently, precursor identification has been applied to Aircraft Loss of Control in-flight (LOC-I). Lee et al. [59] developed an air traffic simulator so that a trained Auto-Encoder can recognize in real-time patterns in-flight data. Auto-Encoders are particular architectures in deep learning that attempt to reconstruct an input it received. The Auto-Encoder can learn to reconstruct normal flight operations and define a statistical baseline using multivariate Gaussian distribution. Abnormal operations are detected when the reconstruction error is considered significant by the statistical baseline. The authors showed that they could detect an abnormal rate of climb/descent and longitudinal and lateral coordinates after a rudder trigger is initiated within the simulation environment.

Deshmukh et al. [22] proposed a precursor mining algorithm that includes unsupervised and supervised methods. The authors first pass time-series surveillance flight data obtained from LaGuardia Airport containing aircraft states through the Temporal logic

learning-based Anomaly Detection (TempAD) [60]. TempAD is an unsupervised anomaly algorithm capable of generating nominal bounds of normal operations in terms of time and features. The outputs from TempAD are time-series with labels, which then go to a supervised precursor detection model that predicts the precursors to each the previously detected anomaly. For the precursor detection model, the authors tried an ANN for both go-around and S-turn anomalies and an SVM model for the go-around one. The ANN led to less false positive a hence performed better.

2.3.4 Literature Review Summary

Over the past years, the application of advanced analytics aiming at improving aviation has tremendously increased. Multiple use cases have been developed; however, most of them relate to anomaly detection and predictive maintenance, which is highly related to system health monitoring. Usually, the methods presented for these use cases tend to be prognostic since anomalies or degradation of the system are detected when they occur, limiting the reaction time for potential corrective actions. Recent work aims to develop more predictive capabilities which involve precursor mining. Table 2.3 summaries the pros and cons of each of the methodologies discussed in subsection 2.3.3.

All the methods mentioned can identify precursors; however, the techniques used are different. Most of the techniques [20, 21, 22] require labeled data and do not use models with an inherent temporal understanding of data such as RNNs, which might leave out critical information present in the time-series. Though ADOPT [17] captures this temporal information, its ability to detect precursors is limited by the required sensitivity analysis. Indeed, the model requires each parameter to be individually perturbed and the features generating the most significant perturbations are considered precursors. The method, therefore, lacks the capability of discovering interactions among precursors. Lee et al. methodology can flag the moment in time where an upset rudder is triggered; however, the detected event does not fall under the used precursor definition [17, 22, 20], as the anomaly

Table 2.3: Precursor Mining Methods Advantages and Disadvantages

Authors	**Advantages**	**Disadvantages**
Janakiraman et al. [15]	• Temporal pattern discovery • Identification of precursor and precursor's region of time	• Sensitivity analysis to identify precursor • Inability to inherently capture interactions between precursors • Requires label of adverse event
Ackley et al [20]	• Precursor identification using interpretable model • Progressive change of precursor at fixed snapshot	• Model inherently does not capture temporal information • Multiple models are required to be trained • Requires label of adverse event
Lee et al [59]	• Detection of upsetting event • Real-time detection implementation via developed simulation environment • No labels required	• Model detect event as soon as it occurs (no predictive capability) • Closer to anomaly detection • Model inherently does not capture temporal information
Deshmukh et al [22]	• No required labels • Identification of precursor and precursor's region of time • Interpretable signal temporal logic models	• Precursor models used do not inherently capture temporal information • Precursor model relies on accurately detected anomalies • Search for precursors feature, iterative process, extra feature engineering to achieve best performances

is not predicted. Deshmukh et al. innovate by including an anomaly detection component in their precursor algorithm. Including an anomaly detection model however limits the precursor model capabilities to the accurately identified anomalies. In other words, the precursor model highly depends on the anomaly detection model performances. Further-

more, the method requires a search for precursors to identify which parameter yields the best F1 score.

CHAPTER 3

PROBLEM FORMULATION

The previous chapters introduced the importance of safety in the aviation industry to maintain the industry's predicted growth. Several encouraging aviation safety statistics were presented, along with methods that have been developed to keep accident rates to the lowest. Modern methods leverage data availability in the industry to establish new advanced analytical approaches through the usage of tools such as machine and deep learning. A literature review focused on work and studies that benefited from these tools revealed several gaps highlighted in the previous chapter. These gaps and the need for continuous safety improvement in aviation motivated the formulation of the research conducted for this thesis:

Research Objective:

Develop a data-driven methodology that will help expand proactive approaches in aviation, and improve flight safety by enhancing modern algorithms used to identify precursors of adverse events, leveraging the identified precursors to retrieve the potential causes of the events, and ensuring that the developed algorithm will be used in a real-world setting within its limits.

The overall research objective is addressed by developing three research questions. The questions and their tentative answers, namely three hypotheses will be tested via three experiments. The development of the research questions and their hypotheses, and an overview of the experiments are described in this chapter.

3.1 Research Questions and Hypothesis Development

3.1.1 Research Question 1

A great emphasis was placed on precursor mining in the literature review due to its advantages and novelty. However, most of the current work leveraging advanced data mining techniques concentrate on anomaly detection and health-monitoring use cases. Unlike anomaly detection, precursor identification provides forecasting power thanks to its ability to predict adverse events ahead of time. On the other hand, system health monitoring can also provides forecasting capabilities at times, but they are limited to health-related anomalies. Overall, a limited amount of studies involved developing precursor mining techniques. Among the reviewed precursor mining techniques, several gaps were observed:

- Limited usage of algorithms that natively handle temporal data (e.g., RNNs)
- The usage of a bias sensitivity analysis due to the inability to track the contribution of individual parameters and their interactions when neural networks are used
- The lack of forecasting capability

Thus the first research question is as follow:

Research Question 1:

How can current temporal-based data mining methods be improved to identify precursors of adverse events and account for their potential interactions?

To answer this question, it is helpful to look back at each of the individual methods and consider their advantages highlighted in Table 2.3. Both methods developed in [17] and [20] were able to identify precursors to the known pre-defined adverse events, these events being a high speed during approach and an unstable approach, respectively. However, recurrent Neural Networks were only used for the DT-MIL model [17]. RNNs are

particularly useful for sequence-based tasks [61] making them the best candidate for capturing the data's temporal information. However, they are neural networks, it becomes hard to keep track of the importance of all parameters. Indeed, the work done in [17] showed that the model could not provide the most important features without using an additional analysis (i.e. sensitivity analysis) which does not take into account potential interactions of the features. Extending the ability to automatically retrieve the aircraft's parameters that correlate to the adverse event is needed to improve ADOPT's shortcoming. Based on these observations, the following can be hypothesized:

Hypothesis 1: *If deep learning methods that extract temporal information are extended such that individual and combined contributions of aircraft parameters are automatically retrieved, then precursors will be identified without any bias.*

In other words, a carefully chosen deep learning method is hypothesized to be able to identify precursors and to automatically provide them and their potential interactions solely based on its architecture and what the architecture learns from the data. Experiment 1, is therefore designed to test this hypothesis.

Experiment 1: Precursor Identification and Quantification Via Precursor Scores

The overview of the experiment is highlighted in Figure 3.1. The experiment starts by processing the data as highlighted later in section 4.2. A precursor model is then trained to forecast an adverse event of interest (i.e. classify whether or not an event will occur) so that the model learns to correlate aircraft parameters to the event. According to defined metrics, a hyperparameter search is used to determine the optimal training conditions and the optimal model parameters that yield the best model. The search ensures that the model

prediction performances are as accurate as inherently possible.

Similar to ADOPT, the precursor model is composed of an RNN, but it is extended with another type of neural network, a Convolutional Neural Network (CNN). Using multi-head CNNs and RNNs such as GRU, it is hypothesized that it is possible to extract important features and discover temporal patterns by leveraging each of the algorithms' inherent capabilities. Each aircraft parameter is assigned its own set of convolutional layers so that each feature goes through a feature extraction process, which emphasizes the importance of some parameters and reduce the impact of unimportant ones. The extracted feature maps are then concatenated and passed to the GRU, which looks at the important information retrieved by the CNN and determines when this information is the most relevant (i.e. when a precursor lies) in time.

In the experiment, unseen flight data that experienced the studied anomaly is given to the precursor. The output of the CNN and RNN layers are then extracted and used to compute a precursor score. In particular, the CNN layer's output is used to identify precursors, and the output of the GRU layer yields the region of time where the precursors are active. Finally a visualization comparing the identified precursors values during an abnormal event and nominal operation is created.

The data processing, model development, precursors identification, and visualization are all made in Python and leverage open-source libraries pandas, numpy, pytorch, matplotlib and seaborn.

3.1.2 Research Question 2

As previously mentioned in Chapter 2, current processes to understand the causes of an adverse event are manual and not scalable [17] which motivates the need for effective methodologies to explain flights. The literature review highlighted the following gaps:

- Multiple methodologies computed precursor scores [17, 20] but a consistent approach for identifying the underlying causes of events using their precursors is lack-

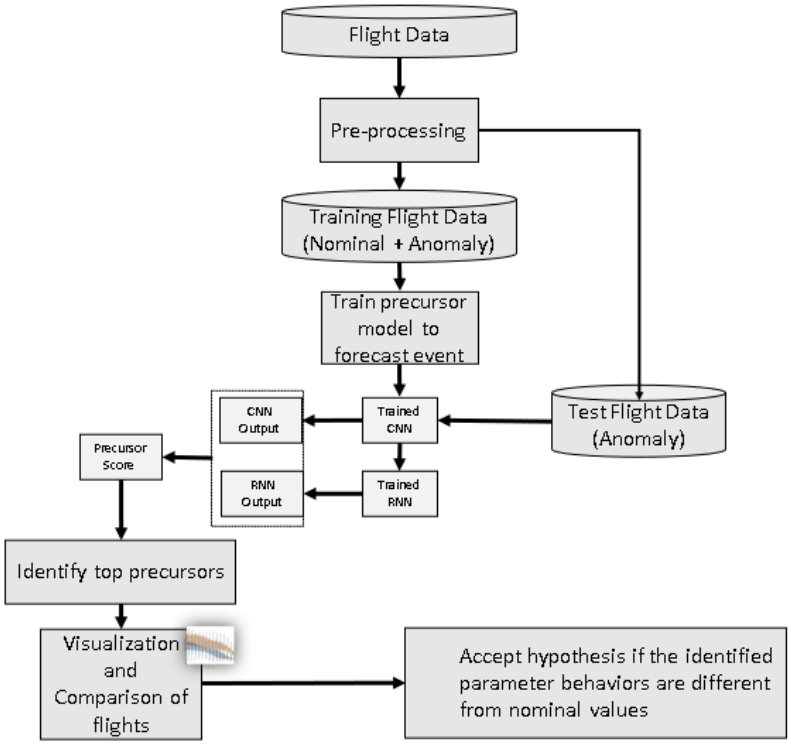


Figure 3.1: Overview of Experiment 1

ing

- Only some of the methodologies presented in the literature actively consider multiple potential causes for the same event

These gaps, therefore, leads to the following research question:

Research Question 2:

How can a standardized approach take advantage of identified precursors to discover potential causes of multiple adverse events?

When precursors are identified, they can be used to understand the event they precede. In [17] and [20], the authors ranked precursors using a score and leveraged it to explain the cause of high-speed events and unstable approach events, respectively. Furthermore, as demonstrated in [22], it is possible to have multiple precursors leading up to the same event. Thus, a formal approach that take advantages of the computed precursor scores of

any precursor mining algorithm to determine multiple potential causes of an adverse event is required, and it is hypothesized that:

Hypothesis 2: *If precursor scores are computed for the identified precursors and used to cluster flights that experienced adverse events, then the clusters will be analyzed to discover potential causes of these events.*

On a fleet-level, grouping flights based on their precursor scores allows for a more granular approach to explaining the causes of adverse events and potentially preventing a dominating abnormal behavior from hiding other non-dominant abnormal patterns.

Experiment 2: Discovering Potential Causes of Adverse Events at a Fleet-Level

The experiment is summarized in Figure 3.2. It is hypothesized that the extracted precursors from **experiment 1**, are highly correlated with the adverse event being studied and therefore can provide insights into the cause of the event. From the identified precursors, precursor scores are created and assigned to each parameter. A table containing precursor scores (precursor score matrix) for each flight parameter is then created, and the data is grouped using the K-means algorithm, an unsupervised learning algorithm.

The algorithm assigned to the same clusters flights that have similar precursor scores. Each cluster can then be analyzed to identify its top precursors. A comparison between the time-series of the original values of the precursors of each cluster can be made through visualizations. Abnormal flights in each cluster are compared to nominal flights. If different abnormal behaviors are observed from one cluster to another, then the hypothesis is accepted. Since different time-series behaviors led to the same event, the different potential causes can be identified.

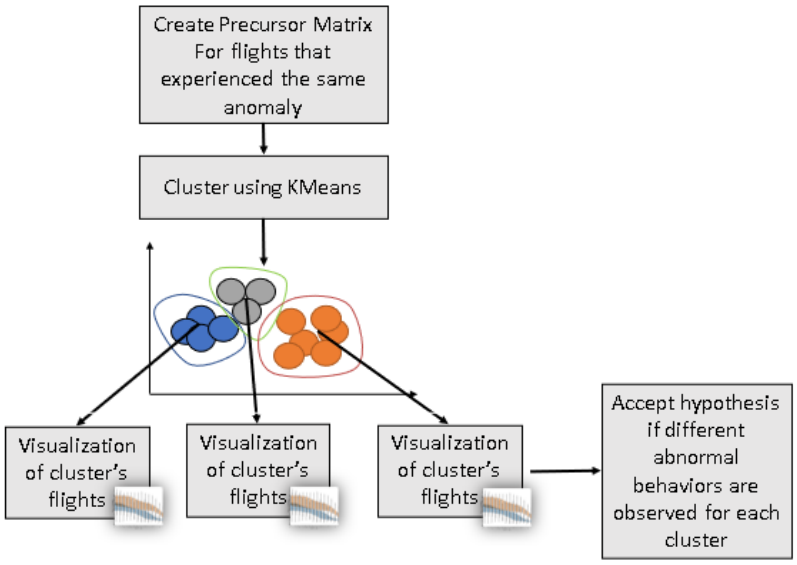


Figure 3.2: Overview of Experiment 2

3.1.3 Research Question 3

Getting access to labeled data is expensive and timely, which is very true for the aviation industry as aviation safety experts must generate the labels. Unfortunately, this leads to a lack of annotated data for the industry. Furthermore, labeled data inherently means that a pre-defined event is known. Having defined known events helps comprehend and learn how to handle them and prevent them from occurring again. However, as aerospace systems are evolving, unknown events could occur due to newly added hardware, software, or new aircraft design. Current techniques would require multiple accidents/events to occur before they could be used to understand them. Finally, predictive models performing classifications tend to wrongly recognize new and unknown classes as one of the known ones, which decreases their performances and usefulness [62]. The gaps observed in chapter 2, can be summarized to the following two:

- Labels are required for most of the techniques
- Supervised learning algorithms operate on a closed-set assumption [63, 62]. The algorithms assume that the training data represent a complete view of the world,

which is not always true

These gaps motivate the following research question:

Research Question 3:
How can the lack of data be compensated for so that the created model's usefulness is ensured?

Deshmukh et al. [22] combined both unsupervised and supervised learning in their methodology. In their work, the authors presented an interesting approach to finding precursors of events that were not pre-defined, since the labels for the precursor model were obtained from an unsupervised model. In addition, the literature reviewed showed that unsupervised anomaly detection algorithms have been leveraged to detect new abnormal flight patterns. From these observations, it can be hypothesized that:

Hypothesis 3: *If an anomaly/novelty detection algorithm is used to flag new anomalies that were not pre-defined and combined with predictive models that learned to recognize defined anomalies, then the created predictive models will be used within their limits.*

A potential answer to the third research question is to combine a novelty detection algorithm with the previously trained supervised models. The novelty detection model can be run before the predictive model to ensure that the latter received data that reassembles the training data, ensuring that the closed-set assumption is met..

Experiment 3: Detection of Potential Unknown Adverse Events and Enhancement of Precursor Model

To validate this hypothesis, flight data is used to train an Auto-Encoder as seen on Figure 3.3. The Auto-Encoder architecture used for this experiment is borrowed from computer vision and presented on Figure B.2. As described in [62], the model is used for Open Set Recognition (OSR). OSR models have the capacity to recognize unknown class, which

is helpful in the real-world. The model is a Variational Auto-Encoder composed of:

- **Encoder:** CNNs and Linear Layers
- **Decoder:** Transpose CNNs and Linear Layers
- **Known Classifier**
- **Unknown Detector**
- **Ladder Architecture** to allow information interactions between the encoder and the decoder

The algorithm is therefore trained to correctly identify flights as either nominal or a known first anomaly, using its classifier. At the same time, it is also trained to reconstruct the input flight. A reconstruction error is then computed and used to determine if a given input flight belongs to a known class (nominal or a known first anomaly) or an unknown class (second anomaly). Using classical classification metric such as the F1 score, the performance of the model can be assessed. Moreover, the precursor model previously trained is later combined with the novelty algorithm and the performances of their combination are also tested. The hypothesis is validated if during the experiment the novelty algorithm is able to detect novelty and if its combination with the precursor model provides better results when novelty (i.e. a new event never seen by either model) is introduced in the testing data set.

A mapping from research questions to hypotheses and experiments that summarizes this chapter is presented in Figure 3.4. To answer the research question presented in this chapter, several hypotheses were made. The next chapter provides more details on the experiments that are conducted to test these hypotheses.

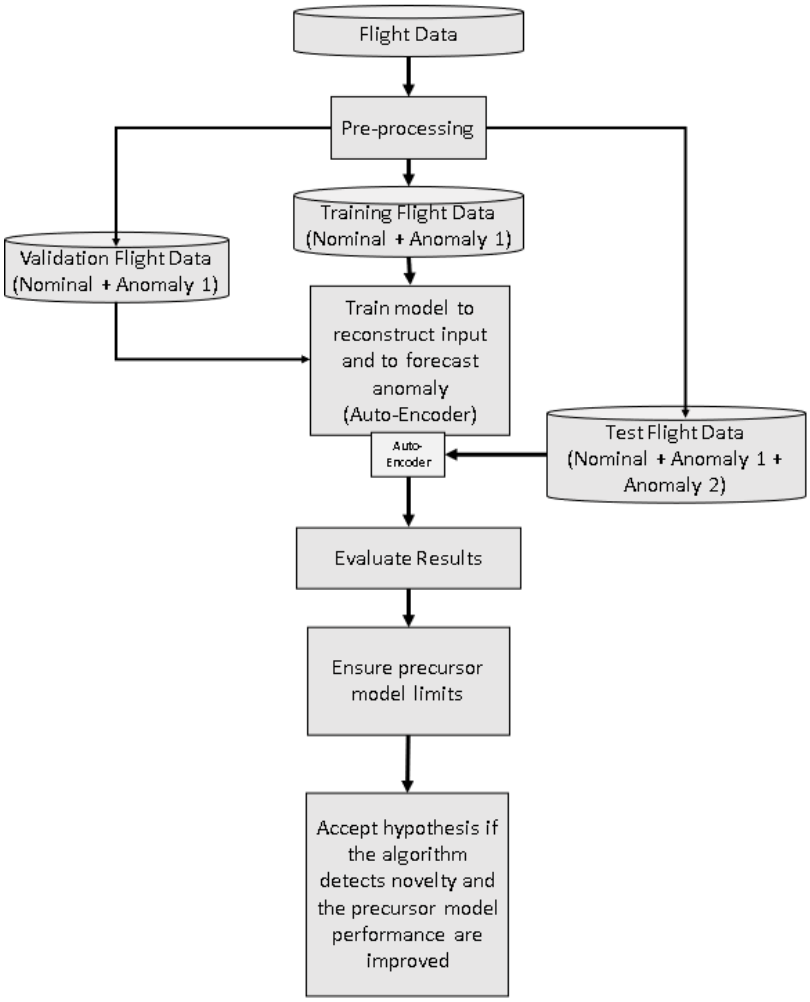


Figure 3.3: Overview of Experiment 3

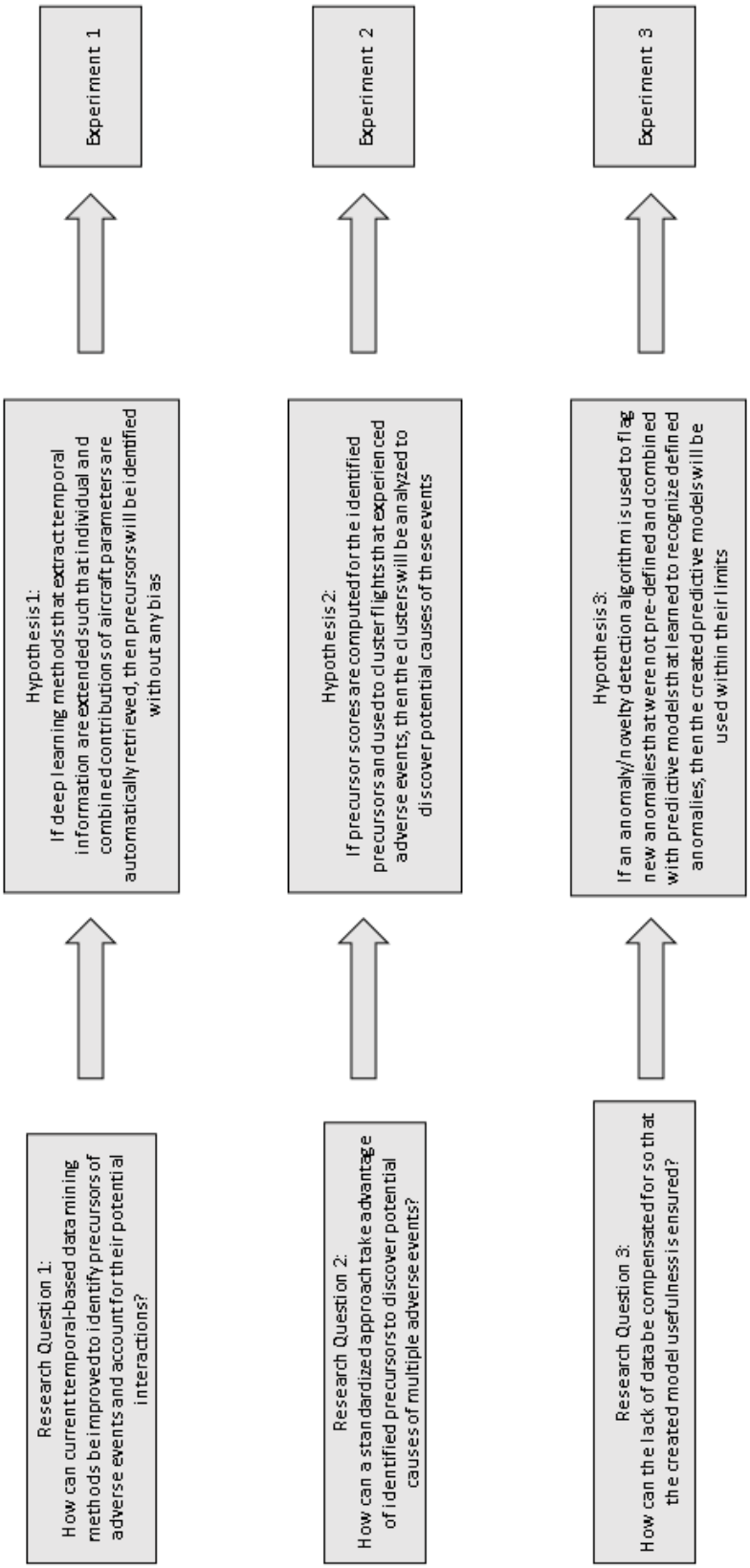


Figure 3.4: Mapping of Research Questions to Hypotheses

3.2 Proposed Methodology

The proposed methodology used to answer this thesis research questions and validate the hypotheses is highlighted in this chapter. As shown in Figure 3.5, flight data are processed using the Intelligent Methodology for the Discovery of Precursors of adverse Events (IM-DoPE). The methodology includes five steps: the data processing, the model development, the extraction of precursors and precursor scores, the flight data analysis, and finally the anomaly detection. The data processing step is common to all hypotheses but steps 2, 3, and 4 are related to **hypothesis 1**, while steps 5 and 6 relates to **hypothesis 2 and 3** respectively. The four outputs are generated by the methodology during the experimental portions of the thesis.

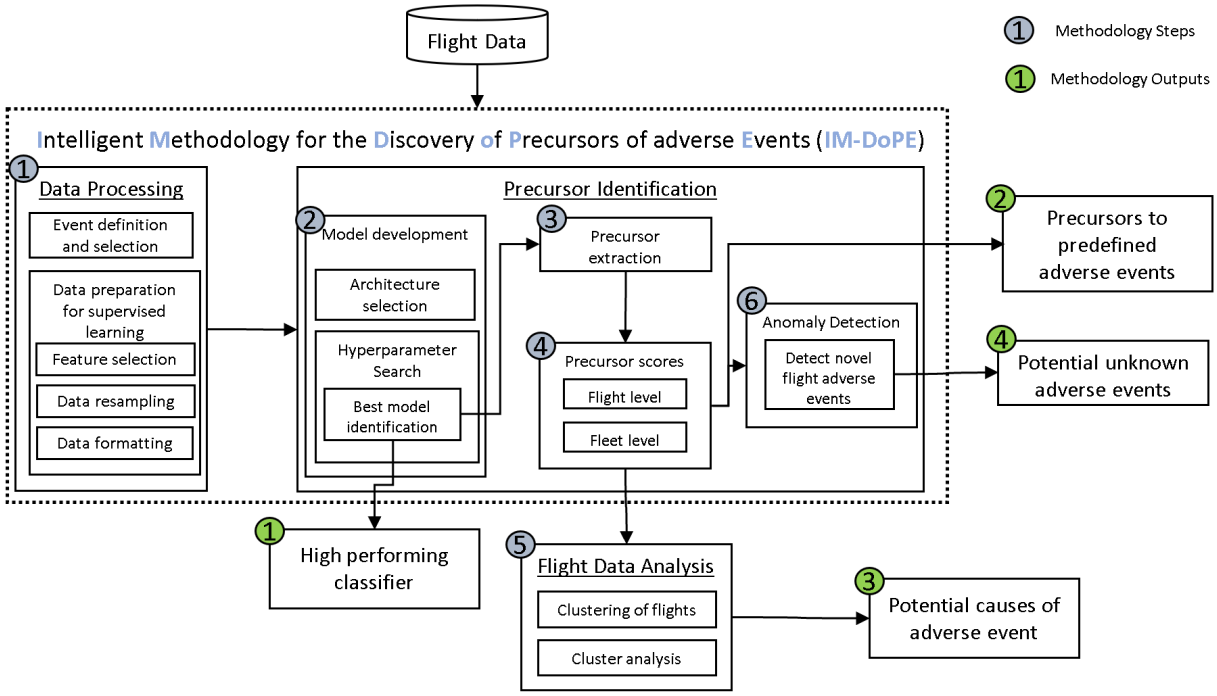


Figure 3.5: Proposed Methodology for Precursor Mining

CHAPTER 4

IDENTIFICATION OF PRECURSORS BY MODEL (RESEARCH QUESTION 1)

Identifying precursors is helpful as it can provide multiple benefits, such as forecasting adverse events and explaining them. The literature noted that current precursor mining methods either lack the inherent capacity to handle temporal data, or the algorithms capable of dealing with temporal data do not provide the contributions of the input aircraft parameters unless extra post-processing steps are performed, such as a sensitivity analysis. These shortcomings led to the formulation of the first research question of this work:

Research Question 1:

How can current temporal-based data mining methods be improved to identify precursors of adverse events and account for their potential interactions?

The DT-MIL framework [17] was the most promising precursor mining method due to its high performances and its inherent capability of dealing with sequential data. However, since the framework uses an RNN, there is a loss of the input parameters' contributions, and the interpretation of precursors becomes difficult. The model required extra post-processing steps (e.g. a sensitivity analysis) to discover precursors. It was therefore hypothesized in subsection 3.1.1 that such algorithm could be improved such that parameters' contributions are better understood. Formally, hypothesis is defined as follow:

Hypothesis 1: *If deep learning methods that extract temporal information are extended such that individual and combined contributions of aircraft parameters are automatically retrieved, then precursors will be identified without any bias.*

This chapter dives deeper into the proposed methodology and highlights the steps required

to get the data ready for a novel deep learning model and how it can extract precursors of flight adverse events.

4.1 Data Acquisition

The methodology is demonstrated using a publicly available dataset obtained from NASA's DASHlink website, a collaborative sharing network for researchers in the Data Mining and Systems Health Management field[1]. The data can be acquired by first downloading a bash script from NASA's DASHlink website, which then can be run to start the entire data set download automatically. In this data set, flight data were recorded for multiple tail numbers of a single type of regional jet operating in commercial service over three years. The data contains detailed aircraft dynamics, system performance, and other engineering parameters but are de-identified such that they cannot be traced back to a particular manufacturer or airline. Since this data set is not part of any airline's FOQA program, additional preprocessing requires creating FOQA-like flags to label individual flights' safety events. The labeling is done by using domain-based rules, similar to the ones presented in [12] though specific parameter values may be used. The defined adverse events are presented in Table 4.1.

Table 4.1: Adverse Events Labeling

Adverse Event	Comments
High Speed in Approach	Flagged at 1,000 ft
Low Speed in Approach	Flagged at 1,000 ft
High Rate Of Descent in Approach	Flagged between 1,000 - 500 ft
High Bank in Approach	Flagged between 1,000 - 400 ft
High Path in Approach	Flagged at 1,000 ft
Low Path in Approach	Flagged at 1,000 ft
Deviation from Localizer	Flagged between 1,000 - 500ft
Deviation below Glideslope	Flagged between 1000 - 500ft
Flaps Late Setting at Landing	None

[1] https://c3.nasa.gov/dashlink/resources/?page=3&sort=-created&type=28

Each of these events is characterized by a severity level ranging from 1 to 3, with 3 being the most severe. For this work, only flights without any safety events and the ones with safety events with a severity level 3 will be considered. Higher severity levels were considered to limit the overlap between nominal and adverse flight operations. Moreover, since it was observed that some events occurred more often than others, the events with higher frequencies were selected:

1. High Speed at Approach
2. High path Angle at Approach

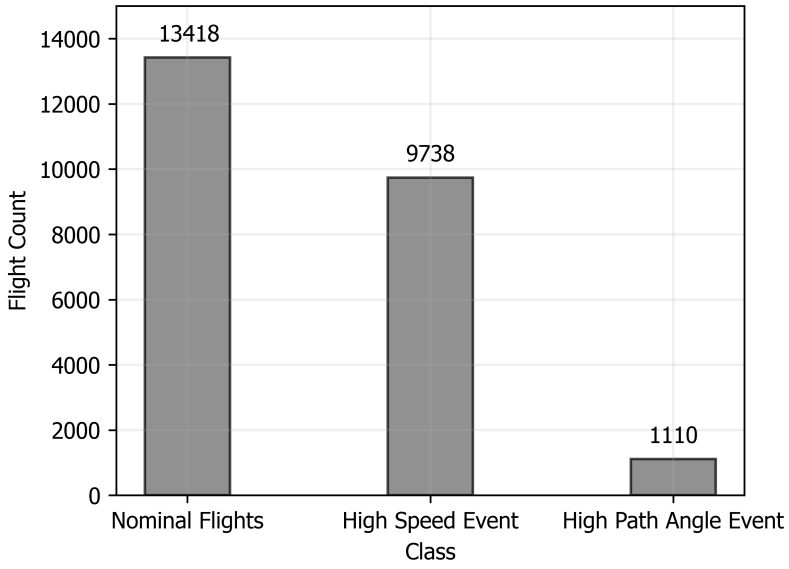


Figure 4.1: Count of Number of Example for each Event and for Normal Operation

As seen on Figure 4.1, the data set is imbalanced (meaning that there are more normal examples than events), which is typical for such problem. The imbalance is managed using stratified mini-batches during the learning process.

4.2 Data Processing

In this section, the preparation of the data for the deep learning model will be explained. These steps are required as the raw data cannot be directly processed by the models due

to diverse reasons such as outliers in the data, its format (flights of different lengths), and considerable variation in scale from one feature to another.

4.2.1 Feature Selection

The outliers in the data were due to bad sensor readings at random time steps and were handled using linear interpolation. Once the data was removed from its outliers, the next step was to down-select the input feature space. Feature selection is commonly made, especially when dealing with linear models due to the curse of dimensionality [64]. The original data set includes about 185 features, both categorical (57) and continuous (128) ones. In such data set, some features may be highly correlated with the target feature. For example, it is expected that speed-related features are the most important when a target feature is a speed-related event. On the other hand, some features might not be correlated with the target feature and therefore bring no helpful information. However, since correlation does not imply causation, the correlation from input features to the target feature was not considered.

The feature selection was based on the correlation between inputs feature themselves. Pairs of features with correlations greater than a set threshold were deemed highly correlated. It is assumed that given information is redundant when two features are highly correlated and that by removing one of them, the input feature feature space can be reduced without any loss of information. Therefore, given a pair of correlated features, one of them can be randomly dropped unless specified that the feature must remain in the data set. If this is the case, the other feature is removed instead. A popular metric used to quantify the correlation among continuous and normally distributed variables is the Pearson Correlation [65]. Usually expressed as a correlation coefficient, this metrics' value is bounded between $-1 < \rho < 1$, where ρ is the correlation coefficient. The value of 1 means that the variables being analyzed have a strong positive association/correlation and a value closer to -1 means strong negative correlation. Finally, a value closer to 0 means week correlation. The

Pearson correlation coefficient for two random variables X and Y is given by:

$$\rho = \frac{\sigma_{XY}}{\sqrt{\sigma_{XX}\sigma_{YY}}} \tag{4.1}$$

Where σ_{XX} is the sample variance for the variable X, and σ_{YY} the one for the variable Y, and σ_{XY} the co-variance between X and Y. Setting a threshold value of $\rho = .90$ resulted in a reduction from 128 continuous values to 67.

Additional non-informative features with constant standard deviation were dropped. The computed airspeed was dropped as it was considered a trivial precursor, particularly for the high-speed event. Finally, except for the N1 Target, parameters related to the auto-pilot (selected altitude, select heading, selected airspeed, and others) were also removed because their inclusion caused the algorithm to identify them as precursors. Although identifying all precursors is essential, identifying multiple precursors of the same type (related to the auto-pilot) provides little information. The final number of features used for the rest of this thesis was 58.

4.2.2 Data Re-sampling

For a given flight, the available parameters are recorded from take-off to landing. Since all the events studied in this thesis are occurring during the flight's approach phase and are flagged at the 1,000 ft above the ground mark, the data is truncated only to contain the last 20 nautical miles away from the 1,000 ft mark. Therefore, the data includes time steps before the actual event, forcing the algorithms to forecast the event. Moreover, each flight is different, and the last 20 nautical miles can correspond to different numbers of data points. For instance, some flights approaching at a higher speed might have fewer points recorded in their last 20 nautical miles than a slower flight. In order to solve this issue and allow for consistency between flights, the data were re-sampled at quarter nautical miles so that each flight consists of about 80 time steps. Additionally, thinking in terms of distance rather

than time is more common to pilots, who can directly benefits from advantages brought by precursors through better training.

The sampling was done for each feature through interpolation using the interp function from the numpy open-source library [2]. The interp function performed a one-dimensional linear interpolation and was found to be effective as shown on Figure 4.2 for the altitude and ground speed parameters.

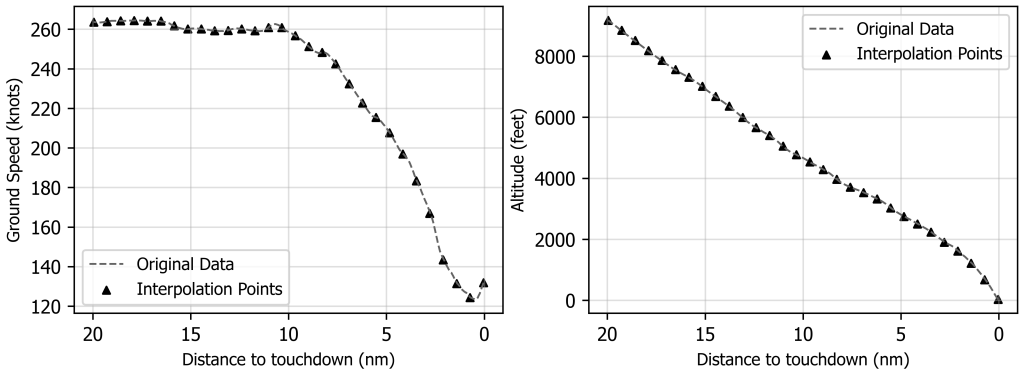


Figure 4.2: Example of Re-sampling on Flight Data

The output of such transformation results in a data table similar to what is shown in Table 4.2.

Table 4.2: Example Results of Interpolating Parameters For a Given Flight

Distance away from 1,000 ft	**Feature 1**	**Feature 2**	...	**Feature d**
20	X	X	...	X
19.75	X	X	...	X
19.5	X	X	...	X
⋮	X	X	...	X
0	X	X	...	X

[2] hhttps://numpy.org

4.3 Data Manipulation

The deep learning architecture that is used for this thesis requires reshaping the data into a tensor. Tensors are multi-dimensional arrays [66] and are particularly useful for this work. Indeed the original data set was reshaped into a 3-Dimensional format, as seen on Figure 4.3. The created array is of dimensions N×L $\times D$, where N is the number of flights, L is the length of each flight, and D is the number of features. For a given flight f, the particular value of a parameter at a given time is defined as $X^f_{i,j}$, with i being the time step i of the flight, and j being the feature or parameter.

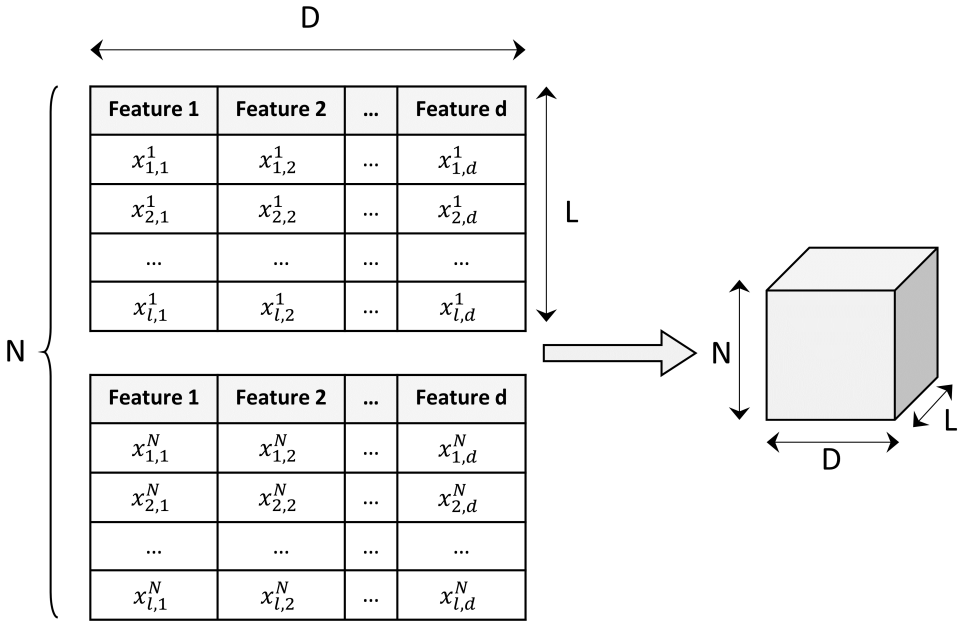


Figure 4.3: Flight Data Reshaping

Reshaping the data in such a way is also helpful for the MIL formulation. Indeed now, each flight f in the tensor is given a label. That is better than specifying an event for each flight's time step, as it might be incorrect. The algorithm can infer the time step labels.

4.4 Precursor Model Development

4.4.1 Architecture Selection

The architecture selection is at the core of this research and is the first step towards testing **hypothesis 1**. Deep learning model architectures can be thought of as combinations of different basic blocks. Two widely used deep learning blocks were selected for this work due to the advantage they provide: the Convolutional Neural Network (CNN) and the Recurrent Neural Network (RNN). The CNN was chosen because of its feature extraction capabilities. In particular, since precursors must be found, a multi-head structure of CNNs was chosen. Therefore, each flight parameter goes through its own set of CNNs, enabling independent, meaningful feature extraction. Indeed, the algorithm learns which aircraft parameters are important and which ones are not, and these parameters can be thought of as the precursors since they are used to forecast an event.

With regards to the sequential aspect of the data, RNNs are better suited for this task. The previously extracted information from the input parameters is sent to the RNN, which finds additional meaningful information related to time, the sequential information of the relevant parameters, and their combinations. For this thesis, a Gated Recurrent Unit (GRU) was selected as the RNN to be implemented as it was successfully used in the DT-MIL framework [17]. This type of neural network requires less time to train and is less complex than other popular RNNs such as the Long Short-Term Memory (LSTM) neural network, which has more parameters.

Finally, the MIL framework is used in conjunction with the two building blocks. After the data goes through the multi-headed convolutions and the GRU, it reaches a time-distributed fully connected neural network. This layer adds more approximation capability to the network [17] and is able to reduce the dimension of the output of the GRU to one abstract dimension. The fully connected layer's output can be bounded between 0 and 1 by using a sigmoid activation function. Each time-step, therefore, has a bounded value,

and max pooling across all of them gives the maximum bounded value for each flight. The flight label can then be obtained by setting a threshold, which would determine if the flight is positive or not. Multiple binary classifiers can therefore be created in this manner, one for each adverse event. The proposed architecture is presented on Figure B.1.

As seen in Figure B.1, multiple convolutions are performed on each input feature. Each aircraft parameter is a univariate time-series. In each head of the multi-head structure, three convolutions acting as feature extractors are performed on the time-series. The architecture increases the number of channels in each convolutional head. That increment can be thought of as the number of dimensions after each convolution. In other words, each of the 1-dimensional time-series gets a dimensional expansion. There is no immediate interpretation of each convolution's additional channels/dimensions, but they are important as they help the neural network learn better from the data. The batch normalization layers after each convolution are commonly used to reduce the internal co-variance shift, bring a regularization effect, and enable faster training [67]. The batch normalization is followed by a Rectified Linear Unit (ReLU), a popular activation function used to improve computational efficiency and minimize the gradient vanishing problem [59]. When the 4^{th} CNN is reached, each head's dimension is reduced back to 1, and a sigmoid activation function is applied. The multi-head structure results are concatenated and can be used to identify precursors and obtain precursor scores, as explained in subsection 4.5.2. This tensor is then sent to the GRU, which has similar hyperparameters to the one developed in [17] and then to a fully connected dense layer. The bounded values of the dense layer's output can be used to retrieve the region of time where the precursors are the most active.

The deep learning architecture presented in this subsection could be used to extract spatial and temporal correlations of aircraft parameters to adverse events, leading to the identification of precursors. The architecture can therefore be used to predict known events such as the ones acquired from the DASHlink dataset.

4.4.2 Hyperparameter Search

Search Setup

Machine learning and deep learning models have model certain parameters that cannot be estimated from the data and must be tuned. These parameters are referred to as hyperparameters and play a role in how the model will perform. Given the novelty of the architecture for precursor mining, the best suit of hyperparameters is not known. Therefore, there is a need to find them through a hyperparameter search. Common strategies to tune these particular parameters are trial and error, grid-search, random search, and Bayesian optimization [68]. This work leverages a grid-search to tune the hyperparameters. A grid search is a naive exhaustive search, and with this approach, each parameter to tune is given an array of values to try. The search is done across all the possible parameter combinations. Figure 4.4 depicts the proposed data split for training the model. For each event, the dataset is first divided in a stratified manner into two sets: train-validation set (80%) and test set (20%). A stratified data split is used such that the proportions of classes (normal and abnormal) are kept in the subsets of the data set. Another stratified split is later performed such that 80% of the original 80% split will actually be used to train the model while the 20% left will be used for the validation set. The validation set is used to optimize the hyperparameters, and the test set is used for the final model evaluation described in subsubsection 4.4.2.

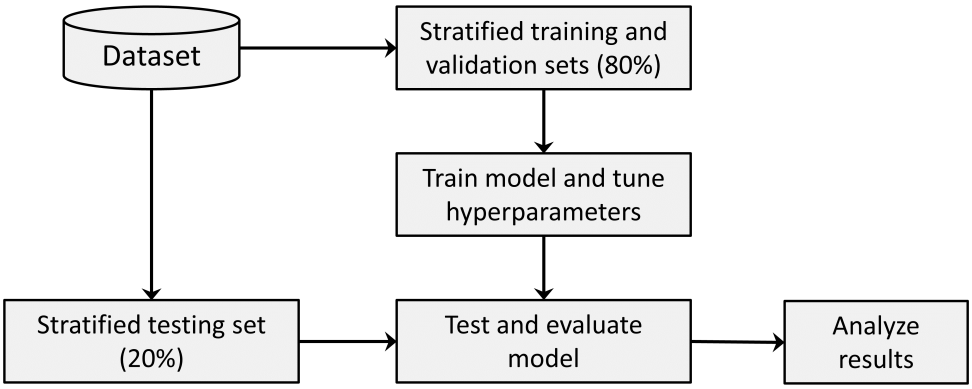


Figure 4.4: Proposed Data Split for Hyperparameter Tuning and Model Training

The list of hyperparameter and the ranges are shown in Table 4.3

Table 4.3: Hyperparaneters and Search Ranges

Hyperparameter	**Search Range**
Convolutional Layers 1,2,3 kernel sizes	[8, 5, 3], [10, 5 ,3], [8, 6, 4], [6, 3, 2]
Convolutional Layers 1,2,3 output channels	[16, 32, 64], [64, 128, 256], [16,48, 144]
Mini Batch-Size Percent	0.1, 0.2
Learning Rate	0.001, 0.0001
Weight Decay (L2 regularization)	0.001, 0.005, 0.01

The first two parameters presented in Table 4.3 are specific to the selected architecture. In particular, the kernel sizes represent the sizes of the filters applied to each time series in each of the convolutional layers. The output channel is the number of dimensions created by that layer for each time series. The last three parameters are model agnostic, but they help improve the algorithm's learning process. The batch size is the number of flights that the model will process at once. After each batch, the model parameters are updated. Finally, the weight decay is the L2 norm that serves the purpose of regularizing the network so that it can generalize to new data.

Best Model Selection

Multiple metrics will be used to evaluate the model performances on the validation set for each of the hyperparameter combinations. Evaluating the classifier models' performances is important as it tells if the model accurately represents the underlying functions that generated the data and what can be expected of the model performances on unseen data belonging to nominal. high-speed or high path events. Several metrics will be used to assess the performances of the trained classifiers:

1. **Confusion Matrix**: Matrix the counts for true positives, true negatives, false positives, and false negatives, as shown by:

Table 4.4: Confusion Matrix

	Predicted: No Event	**Predicted: Event**
Actual: No Event	True Negative (TN)	False Positive (FP)
Actual: Event	False Negative (FN)	True Positive (TP)

2. **Precision**: It measures the proportion of positively labeled examples that are actually positive [32]. The closer to one the better, the following equation defines the precision:

$$\text{precision} = \frac{TP}{TP + FP} \tag{4.2}$$

3. **Recall**: Measures the the fraction of positives labels that were actually detected [32]. The closer to one the better, the following equation defines the recall:

$$\text{recall} = \frac{TP}{TP + FN} \tag{4.3}$$

4. **F1 score**: Harmonic mean of the the precision and recall [32]. The closer to one the better, the following equation defines the recall:

$$\text{F1 score} = \frac{2 \times TP}{2 \times TP + FN + FP} \tag{4.4}$$

5. **Distance from ADOPT (DAF)**: Metric created to measure the resemblance in the precursor score ranking of IM-DoPE and ADOPT [17]. Assuming precursor scores p_i for feature i ranked between 0 and 1, N flights, and d feature, the DFA for each combination of hyperparameters is given by:

$$\text{DFA} = \frac{1}{N} \sum_{j}^{N} \left(\frac{1}{d} \sum_{i}^{d} \left(p_{i,IMDoPE} - p_{i,ADOPT} \right)^2 \right)_j \tag{4.5}$$

The closer to zero the better since the more alike the rankings of the two methods

are. Since ADOPT has been validated for events it predicted by experts [17], it can serve as a second-hand validation if no human expert is available.

The best model can be selected by looking at the highest F1 score archived on the validation dataset, as it would give the best balance between precision and recall. However, when DFA is included as a metric, it becomes harder to select the best model since the model with the best DFA might not be the model with the best F1 score. To mitigate this issue, the best set of hyperparameters and hence the best model are selected using the Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS)[69]. This technique is used for multi-attribute decision making, and a sample of a ranking obtained, and its scoring of combinations is presented in Figure 4.5. TOPSIS chooses the combination that is the closest to the ideal positive solution and furthest from the negative ideal solution. Relative weights are specified for each metric, specifying its importance. In addition to the weights, metrics have to be labeled as cost or benefit. The cost is minimized (DFA), and the benefit is maximized (F1 score). Once the best model is selected, it can be evaluated on the test set.

	learning_rate	l2	kernel_size	n_filters	batch_size_percent	score
Combination_98	0.01	0.005	[8, 5, 3]	[16, 32, 64]	0.5	0.963904
Combination_72	0.01	0.01	[6, 3, 2]	[16, 32, 64]	0.5	0.959570
Combination_122	0.01	0.005	[10, 5, 3]	[16, 32, 64]	0.5	0.958612
Combination_264	0.01	0.001	[6, 3, 2]	[16, 32, 64]	0.5	0.955842
Combination_249	0.01	0.001	[8, 6, 4]	[64, 128, 256]	0.5	0.954031
Combination_74	0.01	0.01	[6, 3, 2]	[16, 32, 64]	0.5	0.951665
Combination_248	0.01	0.001	[8, 6, 4]	[64, 128, 256]	0.5	0.951137
Combination_287	0.01	0.001	[6, 3, 2]	[16, 48, 144]	0.25	0.949720
Combination_146	0.01	0.005	[8, 6, 4]	[16, 32, 64]	0.5	0.948473
Combination_241	0.01	0.001	[8, 6, 4]	[16, 32, 64]	0.5	0.946935
Combination_275	0.01	0.001	[6, 3, 2]	[64, 128, 256]	0.5	0.946719

Figure 4.5: Example of TOPSIS Scoring

4.5 Model Evaluation

4.5.1 Model Evaluation Results

As previously stated, the model's high performances mean that the model learned the underlying dependencies between the input data and the occurrence of an event. It is, therefore, important to know how the model performs. The metrics presented in subsubsection 4.4.2 will be used to evaluate the final model, and its performances will be compared to the one from ADOPT.

4.5.2 Model Interpretation and Precursor Discovery

Once the model is known to perform well, the developed architecture can then be leveraged to retrieve the precursors and test **hypothesis 1**. As seen on Figure 4.5, each aircraft parameter that went through its own layers of convolutions is eventually concatenated back together into a precursor score tensor. This layer can be extracted to retrieve the parameters that the CNNs deemed important and the ones it canceled. Figure 4.6 shows that the parameter radio altitude was critical, making it a precursor, while the rudder pedal positions and the left spoiler were not. Indeed, in that layer, parameters with values of 0.5 are non-important. As the non-important parameters go through the different convolutional layers, their signals get canceled to zero, suggesting that they do not impact the prediction's outcome by much. The zeroed out signal is then passed through the sigmoid activation, and as seen on Equation 4.6 the output of that yields 0.5

$$\sigma(x = 0) = \frac{1}{1 + e^0} = 0.5 \tag{4.6}$$

Therefore to get a more intuitive precursor score, it is necessary to perform a shift to adjust the zero. Additionally, since the CNNs extract important spatial information but not the relevant time information, the precursor value is only relevant within the region of time highlighted by the grey shaded area. This region of time is obtained by looking at the dense

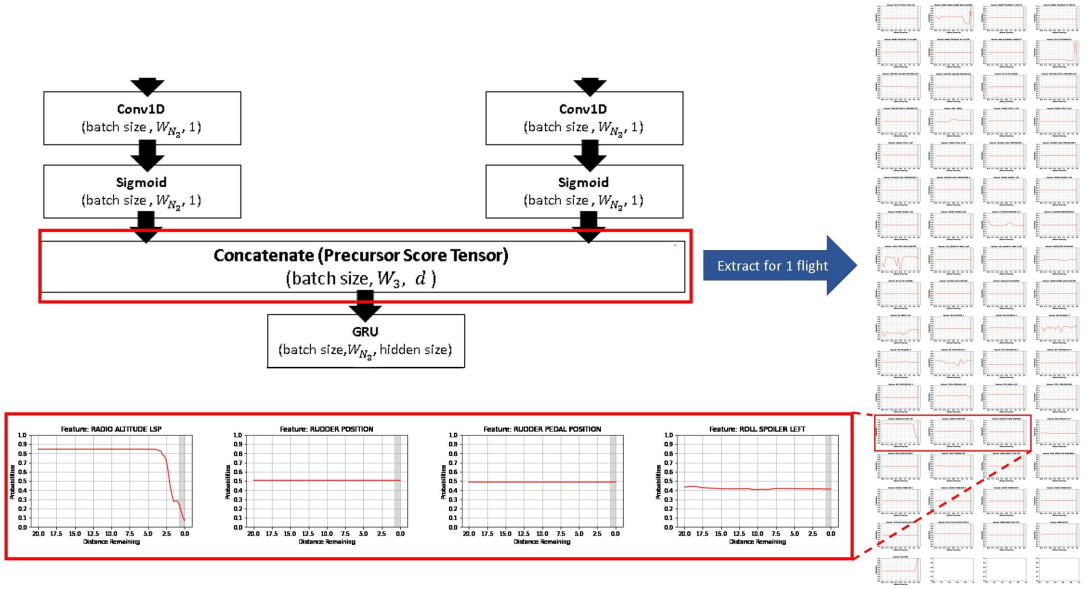


Figure 4.6: Sample Extraction of Precursors from Concatenated Tensor

layer's outputs in the IM-DoPE architecture as suggested by Figure 4.7.

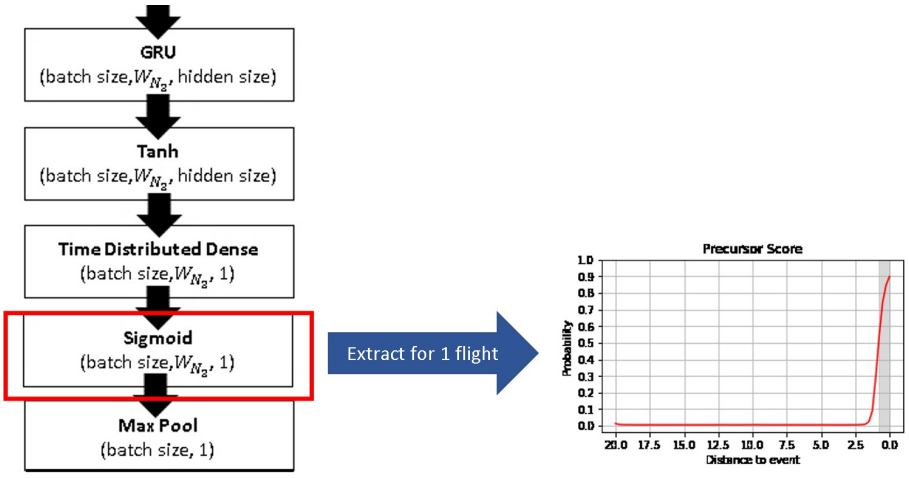


Figure 4.7: Sample Extraction of Precursors Score Over Time from Dense Layer

The adjusted precursor score p_i of feature i is calculated by taking the average of the m shifted precursor scores $p_{i,t}$ where t belongs to the region of time T where the precursors are active, and m represents the number of time-steps which belongs to the region of time. This can be represented by the following equation:

$$p_i = \frac{1}{m} \sum_t \left(|p_{i,t} - 0.5| \right), \forall t \in T \tag{4.7}$$

The expected results of this operation are depicted by Figure 4.8, where the precursors get highlighted while the irrelevant aircraft parameters are set to zero. The wind speed parameter is not relevant, whereas the computed airspeed and altitude are. Another way to present the precursor score is the cumulative precursor score which is just each precursor score p_i divided by the sum of all features' precursor scores. This view allows for the understanding of the contribution of each parameter.

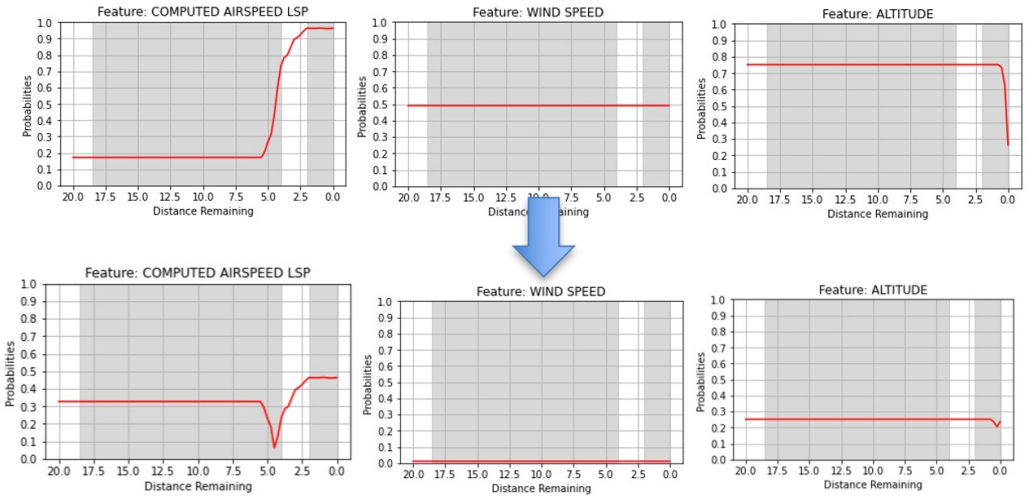


Figure 4.8: Sample Adjustment of Precursor Score

Once the precursors have been identified and the precursor's score has been constructed, the precursor's validation can be done by comparing the results with the identified precursors from ADOPT. Additionally, it is possible to focus on each positive flight's precursors and the region of time they are active. Using this information, the original flight parameters can be retrieved and compared to nominal values using visualization techniques.

4.6 Experiment 1

The steps presented in this chapter are necessary to create a deep learning model that is 1) able to forecast adverse events and 2) determine the precursors to these events. This section demonstrates how the developed model can be used to answer research question 1.

4.6.1 Purpose of Experiment

The proposed experiment is required to test **hypothesis 1** and answer research question 1. This experiment intended to show that:

1. The identified precursors correspond to expected parameters corresponding to a particular type of event and to parameters that ADOPT identified
2. The identified precursors indeed represent parameters with abnormal behavior through comparisons with nominal data

4.6.2 Experiment Setup

For experiment 1, the model is trained on the processed flight data according to the steps highlighted in this chapter. The speed parameter was removed as it was evident that it would be flagged as a precursor. The trained model is used to perform inferences on data from the test set. The true positive flights are then retrieved, and the CNN's outputs are extracted using the methodology outlined in subsection 4.5.2. The features are ranked according to their precursor score averaged across all the true positive flights. The top precursors identified are then identified and plotted using line plots at different time-steps for each adverse event. The line plots allow for a quick and easy way to compare the distributions of the nominal and off-nominal parameters at a different region of time. For both normal and adverse conditions, the median values are shown as dotted lines, and 90% of flights for each case are represented within the shaded regions. The interquartile range represented on these plots represent the difference between the 95th and the 5th percentiles.

4.6.3 Experiment Results

4.6.4 Fleet Level

Overall the final model achieved high performances as seen on Table 4.5. The scores suggest that the model is able to capture the relationship between the input data and the

output.

Table 4.5: Model Evaluation Results

Event	Algorithm	F1 Score	Precision	Recall	DFA
High Speed	IM-DoPE	0.90	0.84	0.97	0.0392
High Path Angle	IM-DoPE	0.83	0.81	0.87	0.0152
High Speed	ADOPT	0.88	0.90	0.86	N/A
High Path Angle	ADOPT	0.70	0.56	0.90	N/A

However, it is important to note that better results for the classical classification metrics are obtained for the high-speed event due to the higher number of training examples and the smaller imbalance in the classes. On the other hand, the larger DFA metric for the high-speed event suggests a greater difference with this event's ADOPT feature ranking. As previously mentioned, the final model is expected to predict adverse events and identify their precursors. The adjusted precursor score can be obtained for each feature and each flight by using the methodology explained in subsection 1.4.1. The average of the precursor scores for the flights that were correctly classified is reported in Table 4.6. The algorithm identified different precursors for the two types of events, which was expected. For example, the glideslope deviation and the pitch angle are characteristics of a high path angle event, while the N1 target relates to engine power related to speed. Some resemblances are also observed. For instance, the altitude is seen to be the precursor for both events. This is expected since the altitude above touch-down is used to define both events, which are both flagged at the 1,000 ft mark. In addition to yielding expected precursors, the model was in accordance with ADOPT, which identified the top 5 precursors for a high-speed event to be the altitude, the radio altitude, the flight path acceleration, and the N1 target. On the other hand, ADOPT identified the glideslope deviation, the pitch angle, the radio altitude, the airbrake position, and the flight path acceleration as precursors to the high path angle event.

Moreover, Figure 4.9 is the visualization of the original flight data for the features

Table 4.6: Average Adjusted Precursor Scores for High Speed and High Path Angle Events

Precursor Rank	**Average Adjusted Precursor Score**	
	High Speed Event	High Path Angle Event
#1	Altitude: 0.31	Radio Altitude:0.31
#2	Radio Altitude : 0.28	Glideslope Deviation: 0.24
#3	N1 Target: 0.25	Pitch Angle: 0.21
#4	Body Longitudinal Acceleration: 0.23	Altitude: 0.18
#5	Lateral Acceleration: 0.14	Flight Path Acceleration: 0.12

flagged as the high-speed event precursors. The altitude features tend to have higher values than the nominal ones. A significant difference can be seen between the medians of nominal and adverse body longitudinal accelerations, notably around the last miles before reaching the 1,000 ft mark. A more significant interquartile range for that feature's adverse flights is also observable. The visualization also shows that the N1 target for adverse is lower than nominal flights until the last 10 miles where the adverse N1 target median remains at zero. Finally, the median lateral acceleration of adverse flights seems to resemble its nominal counterpart. However, a higher interquartile range for the adverse flights is also observed towards the end of the flights.

A very large standard deviations is observed for the high path angle event for the adverse altitude parameters, especially earlier in the approach. The medians of the parameters are much higher for adverse flights, though they slowly becomes closer to the nominal one as the 1,000 ft mark is approached. It can be seen on Figure 4.10 that the glideslope deviation of adverse flights is very different from the one of nominal flights. For the pitch angle, the third identified precursor, the visualization shows a difference in the medians of the adverse flights and the nominal ones starting at about 15 nautical miles away from the event. In addition, a larger interquartile range for the adverse flights is observed. Finally, the flight path acceleration is observed to have more negative values and larger interquartile range for the adverse flights.

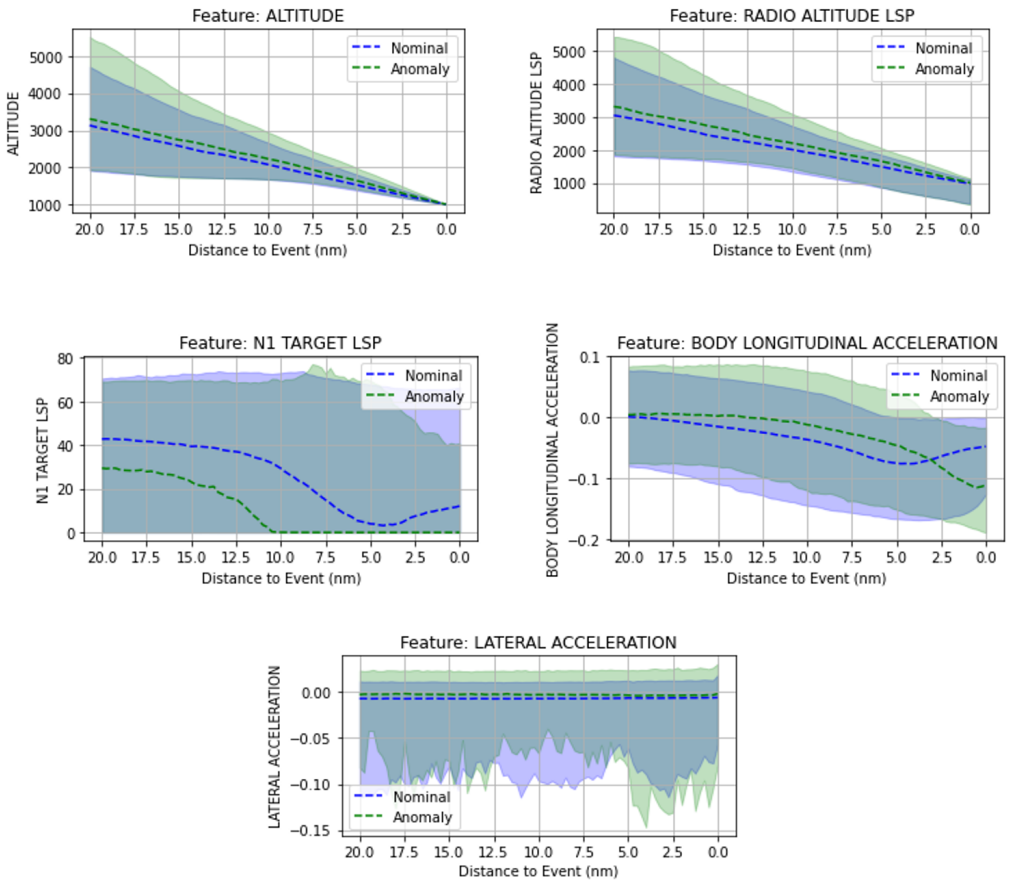


Figure 4.9: Flights Line Plot of Identified Precursors for High Speed Event

4.6.5 Flight Level

The granularity can be increased such that the behavior of the identified precursors can be observed for individual flights. This subsection verifies that the identified precursors indeed displayed abnormalities using representative flights for each adverse event.

High Speed Event: For this flight, the top 5 precursors corresponded to the altitude, the body longitudinal acceleration, the radio altitude, the N1 target, and the total pressure. Notice that the order of importance is different from the fleet's precursor ranking. The ranking of the top 10 precursors of the flight is shown on Figure 4.11. After identifying precursors, they can be plotted as seen on Figure 4.12. The first tile on the plot shows the precursor score over time, which identifies precursor activities. For this flight, precursors

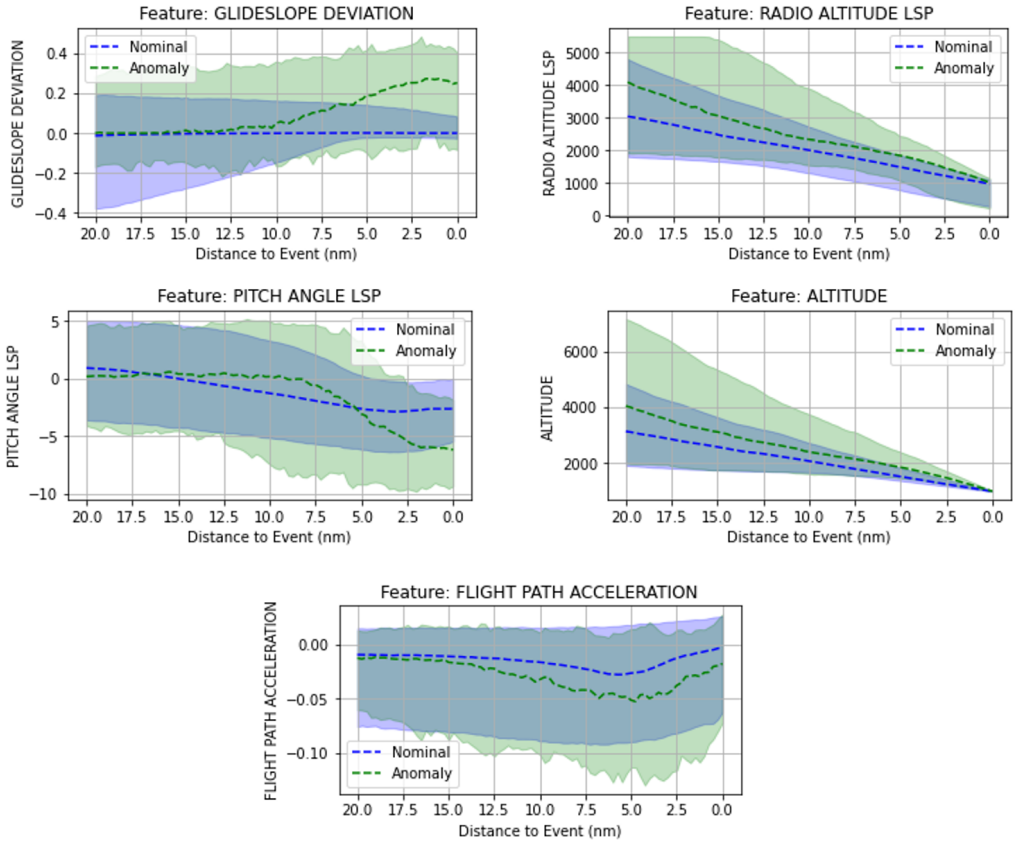


Figure 4.10: Flights Line Plot of Identified Precursors for High Path Angle Event

were active towards the end of the 20 nautical miles leg. The other tiles of the plot show top precursors. The dashed blue line represents the parameter values for that flight, and the dashed black line represents the mean values for that parameter taken across all nominal flights. The purple area is defined to be ± 2 standard deviations away from the mean values. It can be easily seen that the identified precursors are very different from their nominal counterparts from the plots. Features like the body longitudinal acceleration, total pressure, and pitch angle crossed the shaded purple area, meaning that the values are far from the mean of the nominal flights.

High Path Angle Event Like the high-speed event case, the precursors to the model's high path angle event can be extracted. Again, the identified aircraft parameters are assessed by using visualization. For this flight, a dominating precursor is observed. Indeed,

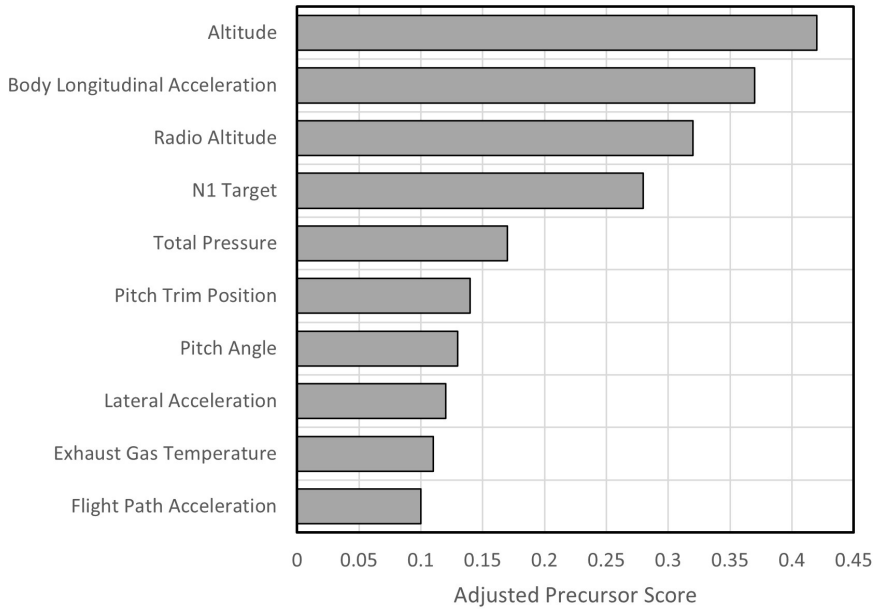


Figure 4.11: Precursor Ranking (High Speed Event)

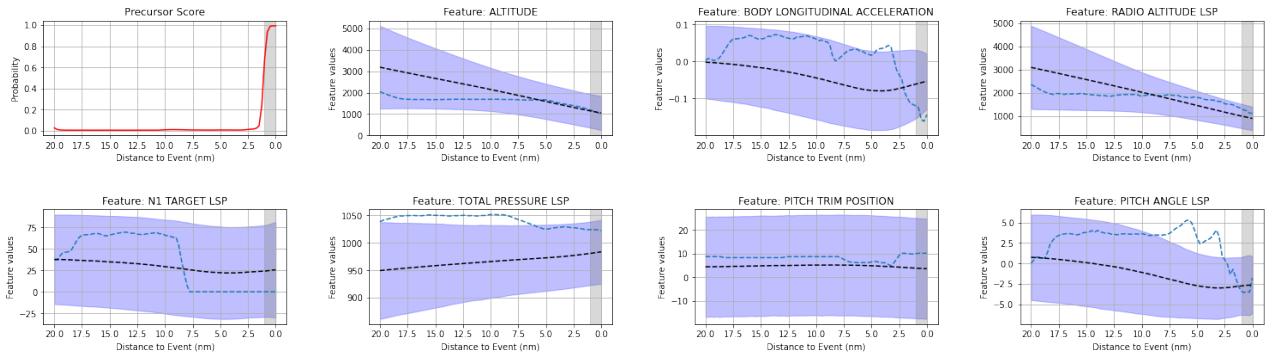


Figure 4.12: Precursor Score and Aircraft's Parameters during a High Speed Event

there is a greater difference between the top two precursor scores than for the high-speed event, as seen in Figure 4.13. This larger difference suggests that the glideslope deviation is highly abnormal. Other important parameters were the pitch angle, the radio altitude, the airbrake position, and the flight path acceleration. The deviant behavior is confirmed by Figure 4.14 since the glideslope deviation is much more significant than two times the standard deviation. Other precursors such as the pitch angle and the flight path acceleration are also identified and observed to have abnormal patterns. In addition, the pitch angle is also a precursor of the high-speed event, as previously observed.

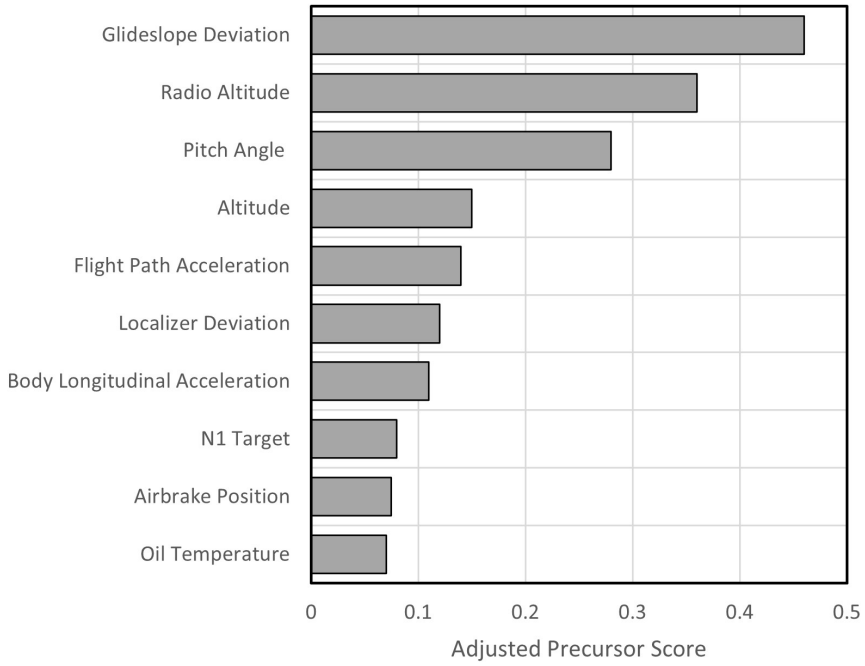


Figure 4.13: Precursor Ranking (High Path Angle Event)

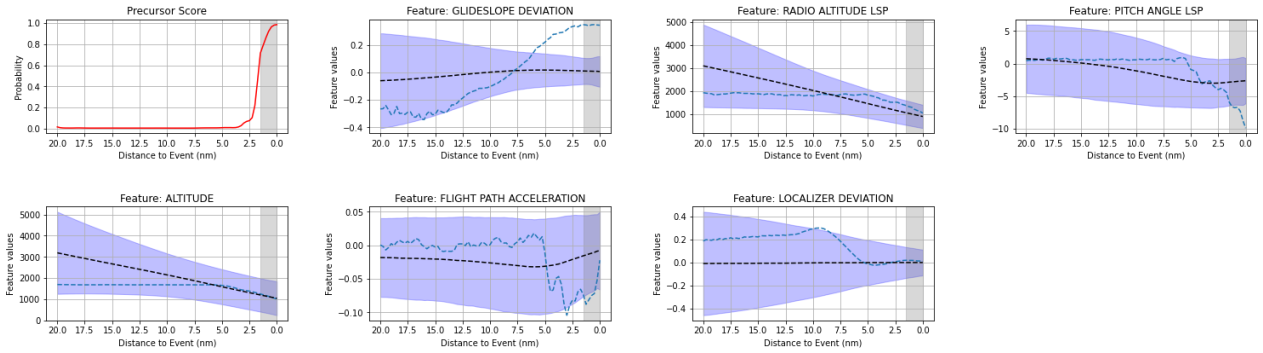


Figure 4.14: Precursor Score and Aircraft's Parameters during a High Path Angle Event

4.6.6 Discussion

Overall the model performances were satisfying as they could forecast a great number of adverse events, and the results obtained were comparable to current similar algorithms (i.e. ADOPT). The high performances ensured that the models were able to capture the relationship between the provided inputs and the output (whether an event occurs or not). That is important since the feature or parameters that helped the models perform well are the ones that mattered the most and, therefore, the ones that relate to the event. This was

in fact checked by looking at features identified as precursors by the algorithm. These features, which were different from one event to the other, were then compared to the top ones observed by ADOPT, which had similar precursors. Finally, visualizations were used to confirm the identified precursors' abnormal behavior by comparing the distributions at every time-step of nominal and adverse flights. These comparisons showed that the identified precursors indeed had different patterns. The results obtained, therefore, support **hypothesis 1** and therefore provide an answer to research question 1,

CHAPTER 5

USE OF PRECURSORS TO EXPLAIN POTENTIAL CAUSES OF ADVERSE EVENTS (RESEARCH QUESTION 2)

The previous chapter highlighted a methodology that takes flight data as input and outputs the forecast of a safety event and any identified precursors to that event. The methodology offers the capability to extract individual flights' precursors and determine the most important precursors over a fleet. However, as seen in the literature [22], multiple precursors can succeed each other and lead to an event. Additionally, since the model in the IM-DoPE methodology enables the extraction of information of individual parameters (CNN) and their combinations (RNN), it can also happen that the combinations of the identified precursors are more important than the individual precursor. Finally, when the precursors are extracted for the whole fleet, the top fleet-level precursors might hide other active precursors, resulting in missing other potential causes of an event. These observations led to the second research question:

Research Question 2:

How can a standardized approach take advantage of identified precursors to discover potential causes of multiple adverse events?

Overall, the developed methodology performs well at detecting precursors but needs to be augmented to explain the causes of an event better. Furthermore, the augmentation needs to be repeatable for different events and leverage the precursor rankings (via precursor scores) of current precursor mining methods. **Hypothesis 2** was therefore defined as such:

Hypothesis 2: *If precursor scores are computed for the identified precursors and used to cluster flights that experienced adverse events, then the clusters will be analyzed to discover potential causes of these events.*

This chapter describes how an ideal number of clusters is obtained and used to cluster flights and how visualization techniques are used to identify diverse potential causes of an event.

5.1 Flight Data Analysis

5.1.1 Flight Clustering Methodology Overview

The precursor scores can be derived and used to explain events that occurred during a flight. The second part of this research aims to develop a structured methodology to explain events using the precursors extracted from IM-DoPE. The methodology is presented in Figure 5.1.

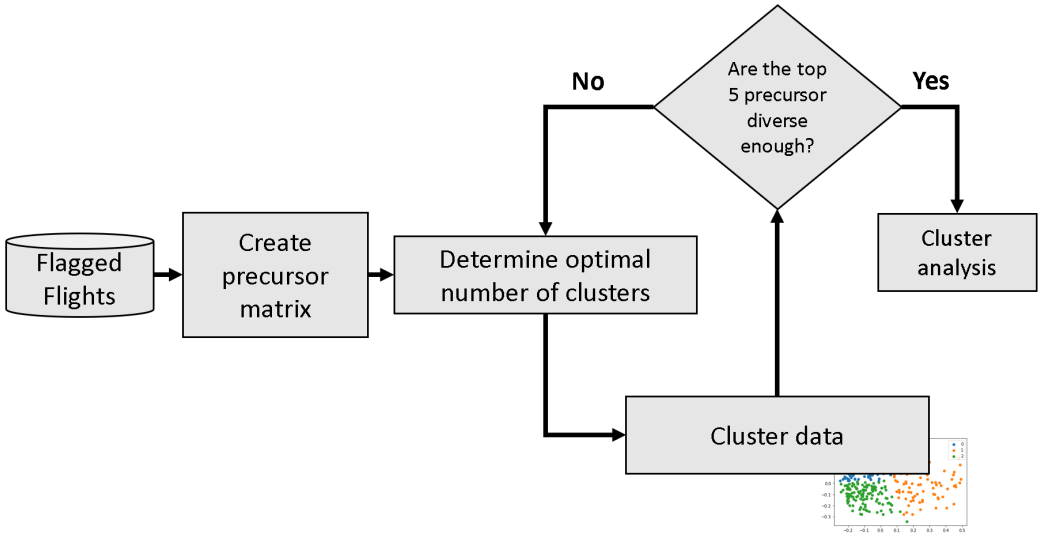


Figure 5.1: Proposed Methodology for Clustering Flights Using Precursor Scores

5.1.2 Create Precursor Matrix

Determining the precursor scores for each feature can be done for flights that are correctly labeled positive by the algorithm, and a matrix of precursors similar to Table 5.1 can be created for each event. Only positive flights are the ones selected as they are since the algorithm correctly predicted the anomaly. The creation of the matrix limits each flight to only its precursors, which enables a different approach to the data analysis instead of directly observing the original values for the parameters of each of the flights. The matrix is next used by the clustering algorithm.

Table 5.1: Sample Precursor Score Matrix

Flights	**Feature 1 precursor score**	**Feature 2 precursor score**	...	**Feature d precursor score**
Flight 1	0.432	0.120	...	0.002
Flight 2	0.492	0.109	...	0.004
⋮	⋮	⋮	...	⋮
Flight N	0.388	0.131	...	0.003

5.1.3 Determine Optimal Number of Clusters

The K-means clustering algorithm can be used to divide the flights into groups. However, the algorithm requires entering the number of clusters before it divides the data as explained in subsubsection A.1. Multiple techniques may be used to determine the optimal number of clusters [70]:

1. **Elbow Method:** The distances of points in a cluster to their respective clusters are squared and summed together. This variance is then plotted for varying numbers of clusters, and the graph initially shows large variances before experiencing a diminishing return effect. That effect can be seen on the plot as an ”elbow,” and the optimal

number of clusters can then be selected.

2. **The average Silhouette Method:** The similarity, known as the silhouette score, of each point to its cluster compared to all the other cluster centers is calculated. This value is bounded between -1 and +1. A higher score means that the point is in accordance with its cluster, while a lower one means that there is a bad match with the cluster. The average silhouette score for each cluster is then computed. If the mean of the cluster is high and closer to 1, then the number of clusters is deemed optimal

3. **Gap Statistics:** The statistics measures the difference between the within-cluster dispersion. It is computed for multiple values of cluster k, and the k value that maximizes the gap statistic is deemed to be the optimal number of cluster

5.1.4 Cluster Analysis

The data set can be clustered into k clusters, with k determined by the previous step. Figure 5.2 shows an example in which the elbow method yielded $k = 3$, and three clusters were created. The dataset dimensionality was reduced to two using the Principal Component Analysis (PCA) [32] for visualization purposes. After initial clustering, the diversity of the created clusters is verified. Since clustering the data is part of unsupervised learning, there are no right and wrong answers. However, some answers are more meaningful than others. Having clusters with different precursors provides more insights than having multiple clusters with the same precursors, which essentially would show only one possible explanation of the event. Verifying that the clusters are diverse is a heuristic process to assess if the top precursors are different enough from cluster to cluster. The top precursors for each cluster are found by taking the average and ranking each feature's precursor scores in each cluster as seen in Figure 5.3. If the top precursors change from cluster to cluster, then the ideal number of clusters is fixed. Otherwise, the next ideal number of clusters is

selected by using the previously mentioned techniques. Figure 5.3 shows the top 5 precur-

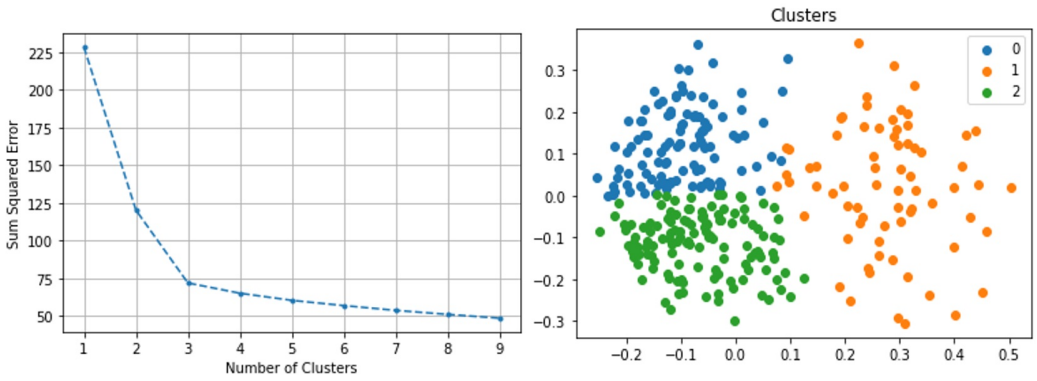


Figure 5.2: Sample Cluster visualization and Elbow Method

sors found in each cluster for a high speed event. In this case, different types of parameters are present in each cluster (parameters related to energy, trajectory, and engine). Cluster 0 and 1 seem more alike though the selected airspeed has a higher precursor score in 0 than 1. This could suggest that an autopilot anomaly could be present, which could also explain cluster 2 that has a high precursor score for the selected course.

Energy related feature (potential)

Energy related feature (kinetic)

Trajectory related feature

Engine related feature

Cluster 0

	SELECTED AIRSPEED	RADIO ALTITUDE LSP	COMPUTED AIRSPEED LSP	ALTITUDE	OIL TEMPERATURE 2
count	1439.00000	1439.00000	1439.00000	1439.00000	1439.00000
mean	0.39919	0.35246	0.34409	0.24868	0.09116
std	0.02123	0.05584	0.06865	0.03248	0.03925
min	0.27492	0.06691	0.05612	0.00398	0.00262
25%	0.40470	0.32383	0.31091	0.24811	0.06167
50%	0.40470	0.32575	0.32342	0.25187	0.09358
75%	0.40470	0.39260	0.40746	0.25187	0.12341
max	0.40470	0.49992	0.47624	0.49914	0.19475

Cluster 1

	RADIO ALTITUDE LSP	COMPUTED AIRSPEED LSP	ALTITUDE	SELECTED AIRSPEED	OIL TEMPERATURE 2
count	1490.00000	1490.00000	1490.00000	1490.00000	1490.00000
mean	0.37334	0.29056	0.25107	0.15422	0.09649
std	0.05656	0.05338	0.04717	0.06130	0.04323
min	0.00036	0.03762	0.01784	0.00198	0.00253
25%	0.33858	0.26255	0.23200	0.10805	0.06632
50%	0.38039	0.32675	0.25187	0.16479	0.10023
75%	0.41471	0.32790	0.26160	0.18629	0.12989
max	0.49586	0.38493	0.47791	0.32682	0.23876

Cluster 2

	SELECTED COURSE	RADIO ALTITUDE LSP	COMPUTED AIRSPEED LSP	ALTITUDE	SELECTED AIRSPEED
count	521.00000	521.00000	521.00000	521.00000	521.00000
mean	0.45065	0.31748	0.29473	0.24961	0.15262
std	0.04898	0.08214	0.05295	0.04290	0.07600
min	0.28330	0.05574	0.01953	0.05022	0.00531
25%	0.45324	0.26914	0.26939	0.23331	0.09558
50%	0.45324	0.32385	0.32484	0.25187	0.16182
75%	0.49076	0.37334	0.32790	0.26278	0.19078
max	0.49549	0.50000	0.44973	0.46065	0.33145

Figure 5.3: Sample Clusters Top Precursors for a High Speed Event

5.2 Experiment 2

This chapter builds on top of the steps outlined in the previous chapter to expand the event explainability capabilities of the IM-DoPE methodology. This section illustrates how identified precursors can be leveraged to explain potential causes of adverse events encountered by flights across the fleet.

5.2.1 Purpose of Experiment

Hypothesis 2 can be tested to answer the second research question. The purpose of this experiment is to demonstrate that:

1. Flights can be grouped into small clusters, and precursors of these smaller groups can provide additional insights not noticeable when identifying precursors of the whole fleet

2. Clusters of precursors are diverse (i.e. they provide different type of information to analysts)

3. Visualization of the identified precursors within each cluster enables the potential explanation of the cause of the event. Different causes are observed as different behaviors are present from one cluster to the other

5.2.2 Experiment Setup

Like experiment 1, true positive flights from the test set are retrieved, and the CNN output for each aircraft parameter is extracted. Using the extracted precursor scores, the precursor score matrix can be created such that an $N \times D$ table is created where N is the number of true positive flights and D is the number of features. The ideal number of clusters k is determined using the elbow method, the silhouette score, and the gap statistics. The K-means algorithm is then used to cluster the flights into k clusters based on their precursor

scores. The top precursors of the k clusters are then identified as they correspond to the parameters with the highest average precursor scores across the flights within each of the clusters. Visualization techniques, namely line plots, are used to observe the abnormal behavior that the precursors and their combination exhibit.

5.2.3 Experiment Results

High Speed Event

Using the precursor scores obtained during experiment 1, the precursor matrix was created, and the data were clustered using K-means. Initially different number of clusters were provided to the algorithm and each of the evaluation method mentioned in subsection 5.1.3 were computed resulting in Figure 5.4.

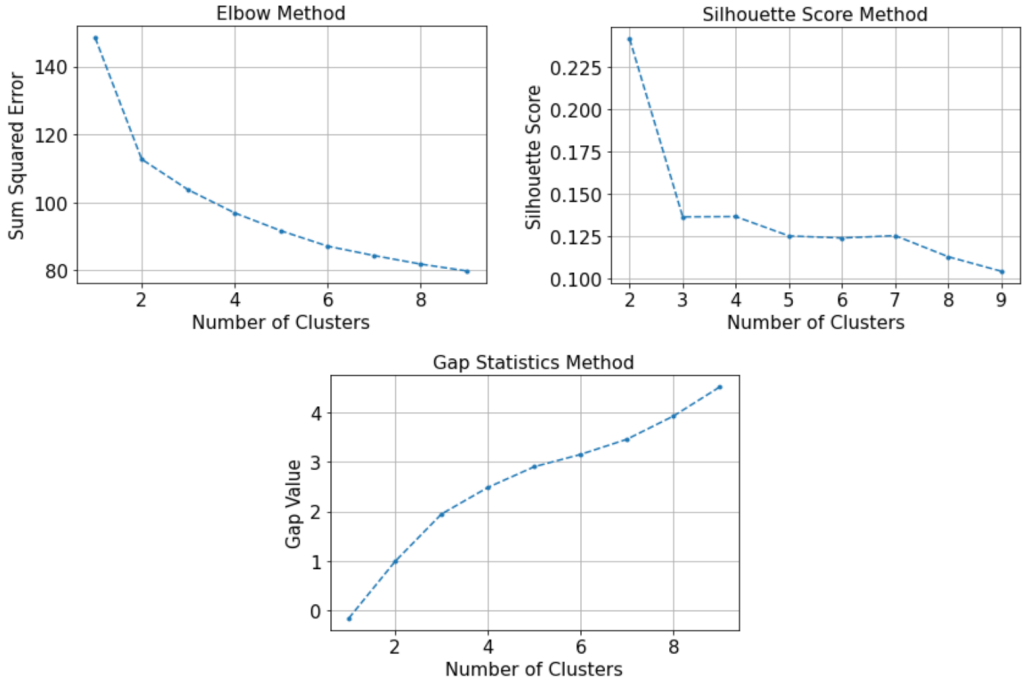


Figure 5.4: Sum Squared Error, Silhouette Score, Gap Statistics V.S. Number of Clusters

From the plots, the elbow and the silhouette score methods are agreeing that the ideal number of clusters is 2 since an ”elbow” is visible at 2 clusters, and the highest silhouette score is obtained for 2 clusters. The gap statistics suggested 9 clusters for this experiment as

the highest value was obtained for that number. Although different from the result obtained from the gap statistics, 2 clusters were deemed ideal.

Cluster 0

	ALTITUDE	RADIO ALTITUDE LSP	BODY LONGITUDINAL ACCELERATION	N1 TARGET LSP	FLIGHT PATH ACCELERATION
count	1085.00000	1085.00000	1085.00000	1085.00000	1085.00000
mean	0.36797	0.32252	0.28562	0.25779	0.15392
std	0.10338	0.05140	0.09508	0.06111	0.05185
min	0.07093	0.05607	0.00105	0.00646	0.00517
25%	0.30816	0.29664	0.22819	0.21318	0.11872
50%	0.38648	0.32591	0.29651	0.29252	0.15561
75%	0.45434	0.35409	0.35737	0.29283	0.18828
max	0.49970	0.48104	0.48709	0.47395	0.35678

Cluster 1

	N1 TARGET LSP	RADIO ALTITUDE LSP	LATERAL ACCELERATION	ALTITUDE	AIRBRAKE POSITION
count	511.00000	511.00000	511.00000	511.00000	511.00000
mean	0.29185	0.27456	0.26692	0.26527	0.24590
std	0.01078	0.08153	0.10220	0.09847	0.05437
min	0.13323	0.00407	0.03120	0.05418	0.00174
25%	0.29278	0.23261	0.19611	0.18534	0.24912
50%	0.29278	0.29105	0.27593	0.28607	0.26172
75%	0.29278	0.32385	0.34906	0.32627	0.27354
max	0.29380	0.43496	0.48248	0.48590	0.28502

Figure 5.5: High Speed Event Clusters

Figure 5.5 shows the top precursors for each of the clusters. The clusters have some precursors in common, but they also have different ones, and the ranking of the common precursors changed from one cluster to the other. The clusters were deemed to be diverse, and therefore, the ideal number of clusters was kept to be 2. The dimensionality of the data is reduced by using PCA, and the clusters are observed on a 2-D plot as seen on Figure 5.6. While the 2-D approximation is able to explain only 43.03% of the variance of the original precursor score matrix, there seems to be a spatial difference for the clusters in the lower representation. The 2-D plot can be used to visualize similar flights, and future flights experiencing high-speed events will lie within either of the spaces.

After clustering the data, an explanation of the anomaly is attempted using line plots. By looking at Figure 5.7 and Figure 5.8, differences can be observed in the patterns for the identified precursors:

- Altitude and Radio Altitude: the average precursor scores for cluster 0 are higher than for cluster 1. The visualization also shows that the medians corresponding to the adverse flights of these parameters, in cluster 0, tend to be greater than the median of the nominal flights

Figure 5.6: High Speed Event Precursor Score Matrix Reduced Dimension

- Body Longitudinal Acceleration: significant differences between adverse flights and nominal flights are observed in cluster 0 hence why the parameter is a precursor of that cluster. The feature is not a precursor in cluster 1, and therefore the adverse flights behaved similarly to the nominal flights as seen on the line plots.

- N1 Target: in cluster 0, the N1 target distribution of adverse flights changes a lot from being similar to nominal to being different. Adverse flights in cluster 1 have a constant zero N1 target, suggesting that the pilots may not be using the auto-pilot.

- Flight Path Acceleration: This feature's behavior is similar to what was observed for the body longitudinal acceleration. The visualization shows more considerable differences in cluster 0, notably when the aircraft is approaching an altitude of 1,000 ft. Cluster 1 behavior is similar to nominal flights.

- Lateral Acceleration: two different behaviors are observed. Cluster 0 reassembles nominal flight, while flights in cluster 1 seem to maintain a constant difference gap between adverse flights and the nominal ones. The lateral acceleration was found to be a precursor only for cluster 1.

- Airbrake Position: abnormal behavior is observed in cluster 0, where a significant

interquartile range is observed for the adverse flights. However, looking closely at the visualization reveals that these flights' median appears to be close to nominal values. Similarly, cluster 1 has close adverse and nominal medians but it has a rapid increase in the interquartile range close to the 1,000 ft mark.

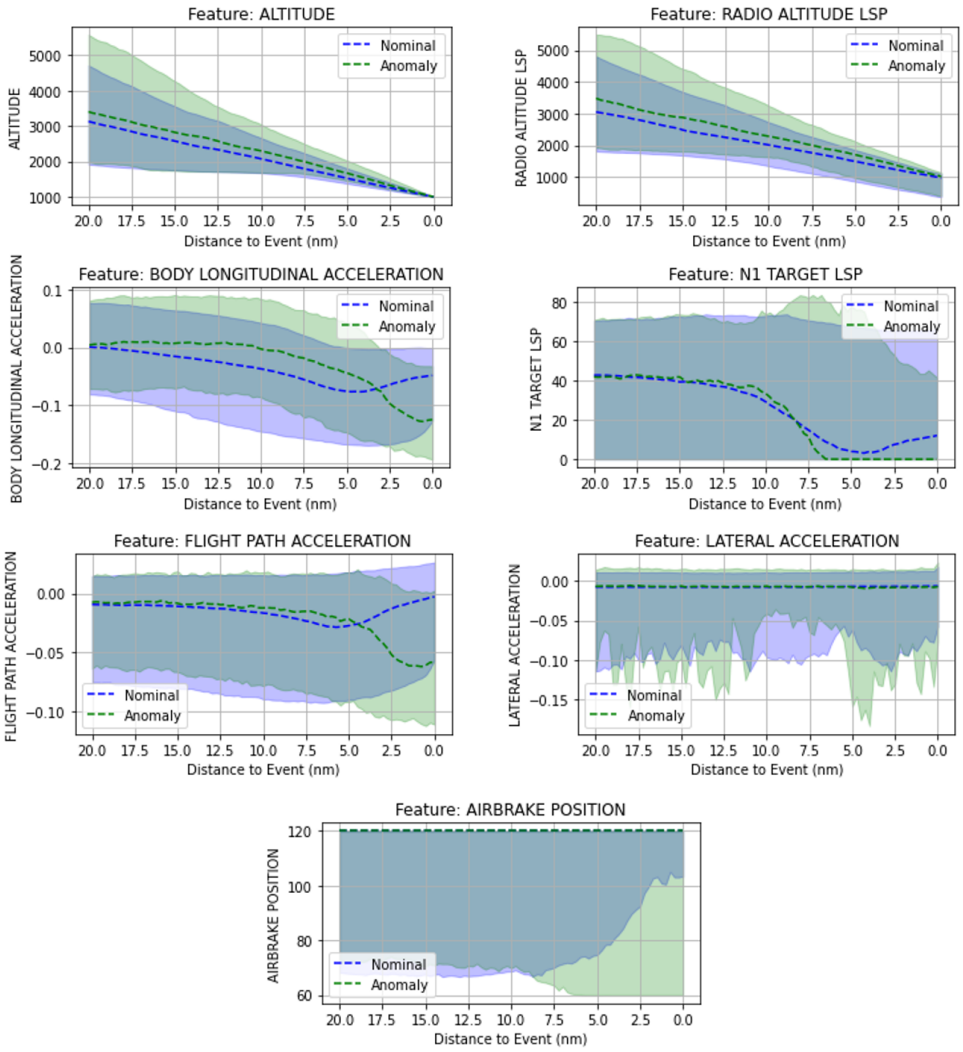


Figure 5.7: Line Plot Comparisions (Cluster 0)

By comparing the values for the two clusters, an attempt to explain the high-speed adverse event is made using two possible causes:

- High-speed event might be caused by the aircraft arriving from a higher than normal altitude (i.e. late descent), which led the pilot to descend faster. The attempt to

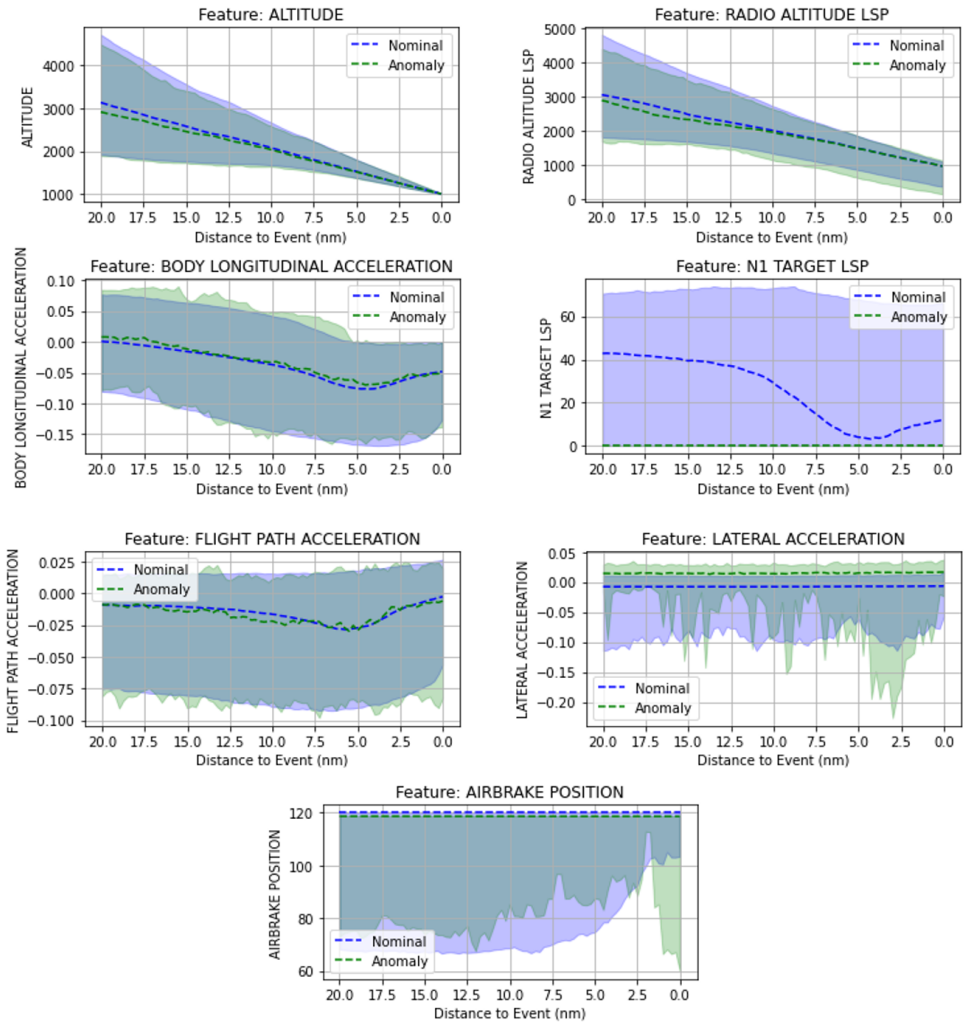


Figure 5.8: Line Plot Comparisons (Cluster 1)

slow the aircraft down is suggested by the negative body longitudinal and the negative flight path accelerations (using conventional axis directions). The pilots seemed to have attempted to started to slow the aircraft down late since the accelerations deviated from nominal in the last 2 nautical miles.

- High-speed event might be due to an issue with the auto-pilot not being set. In fact, the high-speed event is defined by comparing the speed of the aircraft to the selected airspeed on the auto-pilot. Therefore if the auto-pilot is malfunctioning or not set, the flight might be flagged wrongly, and the airline/flight analysis would need to review the labeling process or understand why pilots are not using the auto-pilot.

Additionally, abnormal behaviors are also observed for the aircraft attitude, lateral acceleration, and airbrake.

From observing precursor behavior in the different clusters, similarities and differences can be observed, which helps understand multiple precursor behaviors for a high-speed event. While observations and potential explanations need to be reviewed by flight analysts and experts, this provides a head start into the flight analysis. The same analysis is then performed on a flight with a high path angle event.

High Path Angle Event

Like the high-speed event case, the ideal number of clusters was computed using the elbow method, the silhouette score, and the gap statistics. As shown on Figure 5.9, the ideal number of clusters determined by the elbow and silhouette score was 2, whereas the gap statistics again suggested an ideal cluster of 9. A 2-D representation of the data was also plotted to visualize the space occupied by clusters for the high path angle event, as seen on Figure 5.10. For this event, the 2-Dimensional representation explained 40% of the variance of the original data. However, the separation between the clusters was again visible.

Seven unique features were identified as the top precursors of the two created clusters. The altitude parameters (altitude and radio altitude) were found to be important features. Again this is expected as the high path angle is defined at 1,000 ft, similarly to the high-speed event. Additional features were the pitch angle (which is related to the flight path of aircraft), the total pressure (related to the aircraft's altitude and speed), the localizer deviation, the glideslope deviation, and the flight path acceleration. The clusters are then analyzed by using visualization techniques (Figure 5.12 and Figure 5.13) to gain insights from them:

- Altitude and Radio Altitude: similar abnormal behavior is observed in both clusters, where the parameters for the high path angle flights have larger interquartile range, and the medians are much higher than the ones of their nominal counterparts. These

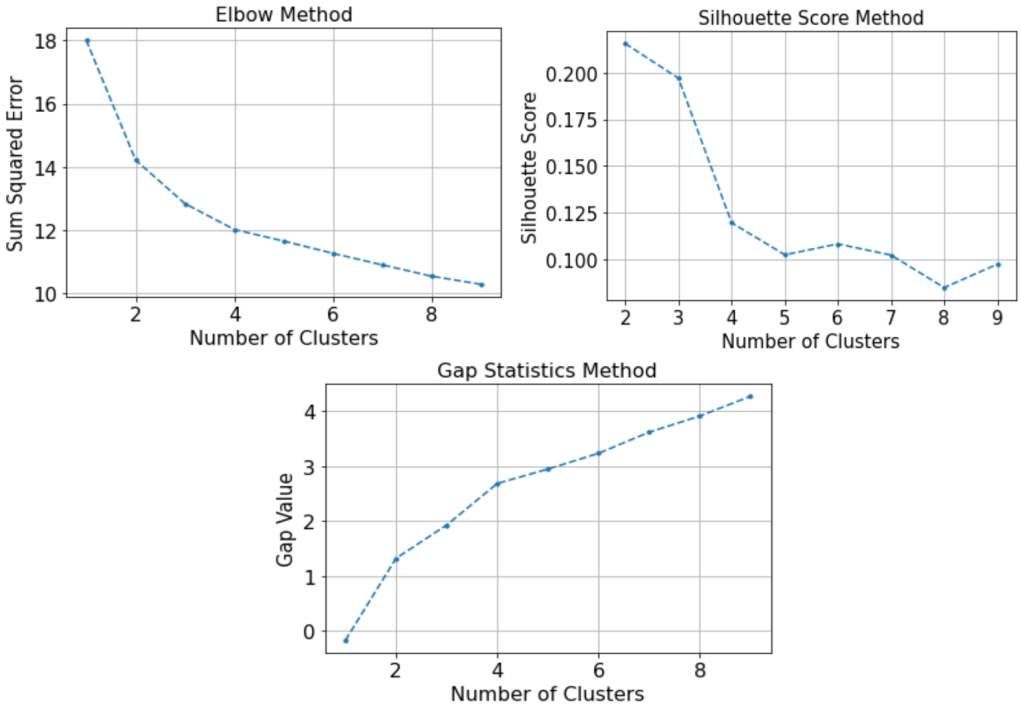


Figure 5.9: Sum Squared Error, Silhouette Score, Gap Statistics V.S. Number of Clusters

larger values tend to occur earlier during the approach. The parameters approach nominal when the aircraft approaches 1,000 ft.

- Pitch Angle: the pitch angle relates to the flight path of an aircraft, it therefore not surprising to see it flagged as a precursor. In both clusters, the pitch angle of adverse flights is lower than the nominal flights before the event.

- Total Pressure: similar behaviors in the two clusters are observed as well. Early in the approach, the total pressure has a higher interquartile range and has a slightly different median than the nominal flights. As aircraft approach the altitude of 1,000 ft, the difference from normal fades.

- Localizer Deviation: in both clusters, significant differences with the nominal flights are observed. For cluster 0, different interquartile ranges at multiple time-steps. A similar behavior is observed for cluster 1.

- Glideslope Deviation: cluster 0 has a large interquartile range for the feature. How-

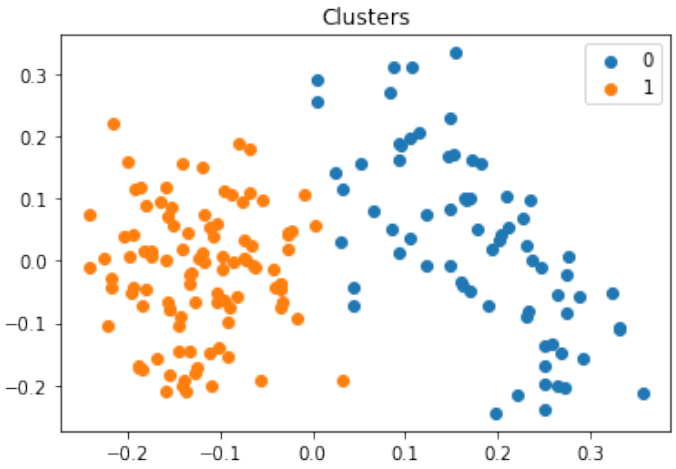


Figure 5.10: High Path Angle Event Precursor Score Matrix Reduced Dimension

Cluster 0

	RADIO ALTITUDE LSP	PITCH ANGLE LSP	ALTITUDE	TOTAL PRESSURE LSP	LOCALIZER DEVIATION
count	70.00000	70.00000	70.00000	70.00000	70.00000
mean	0.36735	0.22612	0.18355	0.16464	0.14205
std	0.09238	0.07215	0.12153	0.10093	0.08663
min	0.08763	0.01835	0.00382	0.01029	0.02263
25%	0.31109	0.19545	0.09440	0.06871	0.09114
50%	0.38640	0.23968	0.17051	0.17001	0.12056
75%	0.44653	0.28545	0.26321	0.26149	0.16817
max	0.49970	0.34796	0.42038	0.32157	0.43441

Cluster 1

	GLIDESLOPE DEVIATION	RADIO ALTITUDE LSP	PITCH ANGLE LSP	ALTITUDE	FLIGHT PATH ACCELERATION
count	102.00000	102.00000	102.00000	102.00000	102.00000
mean	0.37821	0.32061	0.23678	0.22250	0.13429
std	0.06735	0.08370	0.05456	0.08729	0.05690
min	0.20690	0.01582	0.06167	0.02773	0.01489
25%	0.32103	0.28636	0.20989	0.16855	0.09937
50%	0.38543	0.33060	0.24487	0.21517	0.13358
75%	0.43375	0.37562	0.27773	0.27051	0.16409
max	0.49941	0.45847	0.33202	0.44528	0.30328

Figure 5.11: High Path Angle Event Clusters

ever, the behavior of the adverse flights in cluster 1 is very different from the nominal flights. The median glideslope deviation in that cluster is much higher for the adverse flights. The large precursor score for the parameter in that cluster also confirms the visualization.

- Flight Path Acceleration: this parameter was also flagged for the high-speed event, and the behavior of the adverse flights in both clusters is slightly different from nominal values. A smaller acceleration is observed.

By observing the behavior of the adverse flights in each cluster, an attempt to explain what

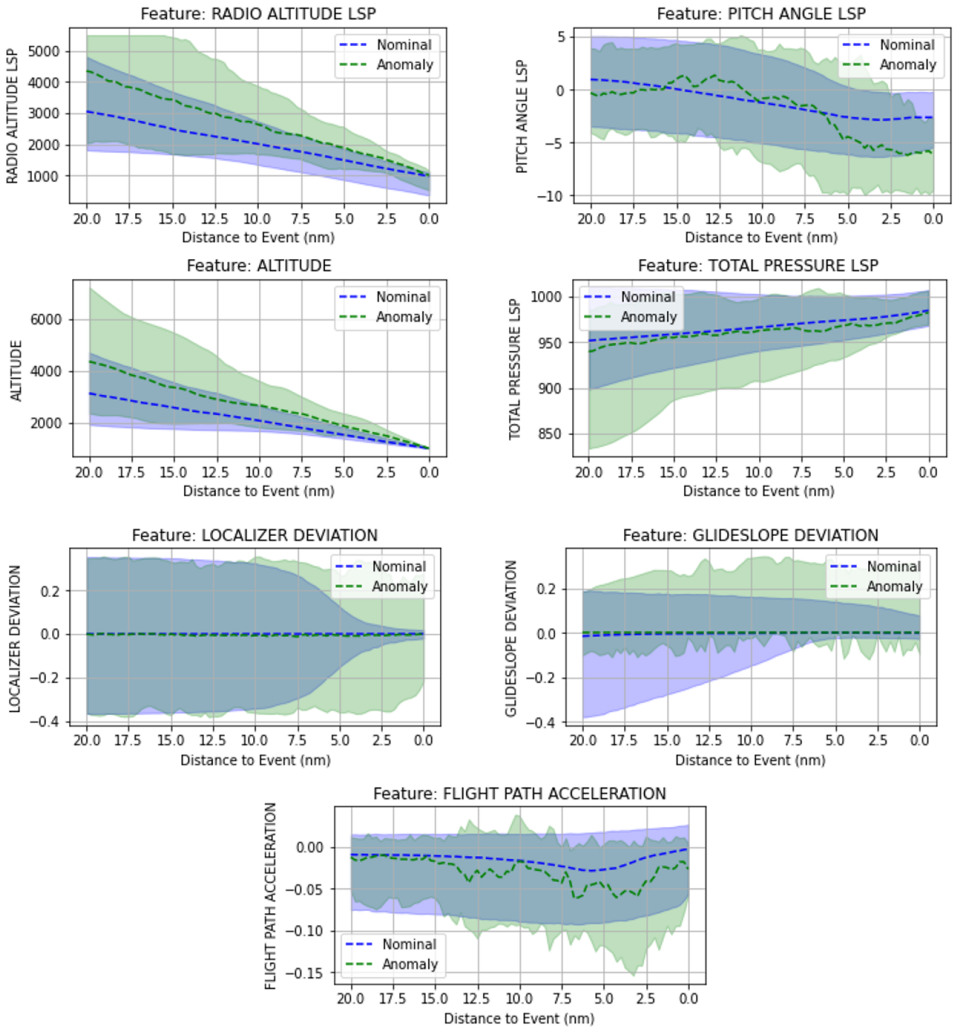


Figure 5.12: Line Plot Comparison (Cluster 0)

could have caused the high path angle events is made:

- Cluster 0 suggests that an aircraft's high altitude early in an approach might lead to a lower than nominal pitch angle at 1,000 ft since the altitude slowly approaches nominal behavior while the pitch angle moves away from it.

- Cluster 1 also suggests a possible low pitch angle due to higher altitude earlier on in the approach. However, a major difference with cluster 0 is the significant glideslope deviation from nominal flights. From the visualization, this cluster seems to have additional underlying causes related to the aircraft's glideslope, for the high path angle event.

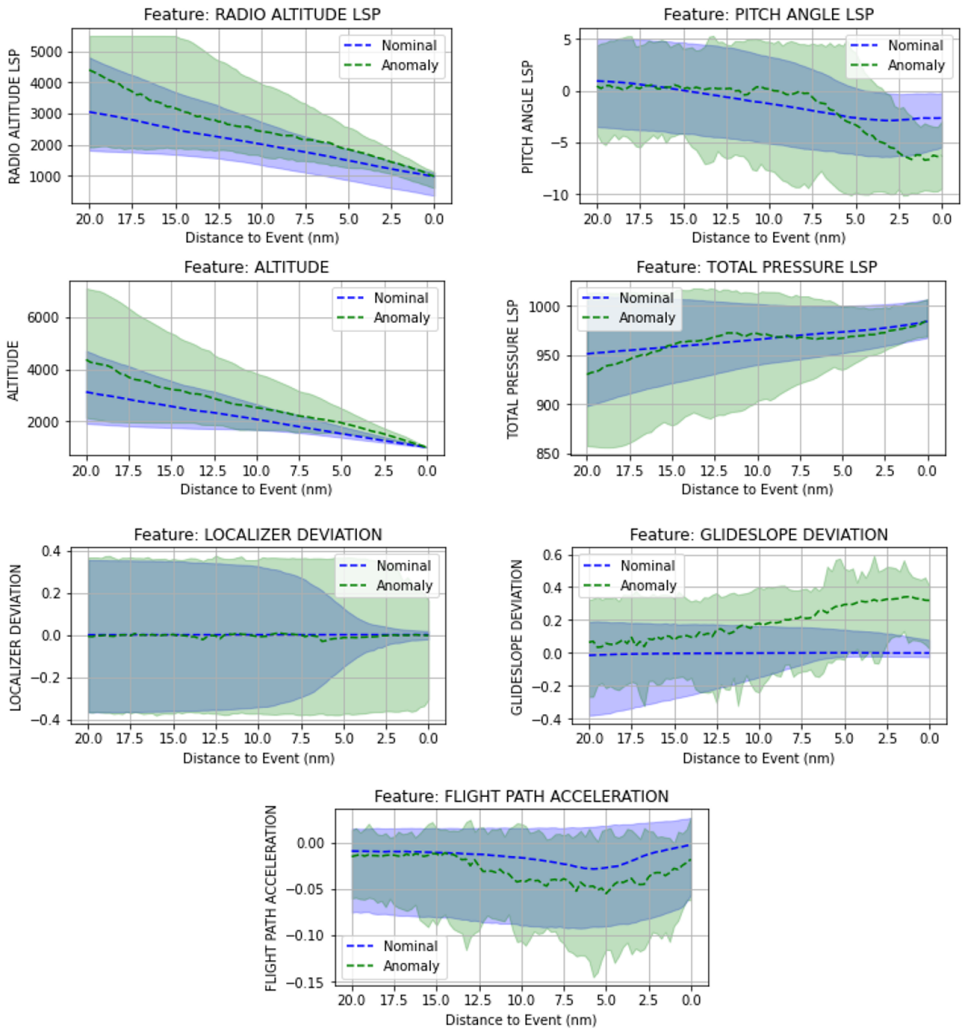


Figure 5.13: Line Plot Comparison (Cluster 1)

5.2.4 Discussion

Clustering is an unsupervised task, meaning that no labels exist, making it hard to understand whether created clusters are appropriate or not. For this thesis, cluster quality measures such as the elbow method, silhouette score, and the gap statistics were used to estimate the ideal number of clusters for each adverse event. Similar results were obtained for both anomalies, where two clusters were deemed to be ideal. It is important to note that because of the high number of clusters suggested by the gap statistics, which continued increasing as in the ideal number of clusters was increased past 9, this measure might not be

appropriate to find ideal clusters for the precursor score matrix. When the cluster's top precursors were retrieved, they were qualitatively deemed diverse as a clear difference could be seen between the cluster's top precursors (i.e. the cluster had either different precursors or their ranking was different from one cluster to the other). The diversity in the clusters already provided insights as it suggested that depending on the situations, some parameters mattered more. Moreover, splitting the flights into groups using their precursor scores allowed for more granularity in explaining the event studied. Different observations could be made from cluster to cluster as the parameters observed behaved differently, suggesting different causes to the events. The visualizations of the identified precursors in each cluster provided the required observation and understanding of the abnormality experienced in the flights. Overall, the results obtained help fulfill the purpose of this experiment, which helps answer research question 2.

CHAPTER 6

PRECURSOR MODEL ENHANCEMENTS WITH NOVELTY DETECTION (RESEARCH QUESTION 3)

As previously mentioned, developed the precursor (predictive) model can be highly confident in its prediction for a new flight even though that flight may not belong to a nominal case or any anomalies that the model has seen during training. Therefore, it is crucial to detect novelty when provided with a new data set such that confidence in the precursor model can remain and new anomalies can be detected. For these reasons, research question 3 was developed:

Research Question 3:

How can the lack of data be compensated for so that the created model's usefulness is ensured?

Anomaly detection models can detect changes within the data distribution without prior knowledge and therefore are good candidates when trying to detect novelty. Thus, it was hypothesized that:

Hypothesis 3: *If an anomaly/novelty detection algorithm is used to flag new anomalies that were not pre-defined and combined with predictive models that learned to recognize defined anomalies, then the created predictive models will be used within their limits.*

This chapter describes the architecture used for the novelty task and along with the results obtained for experiment 3.

6.1 Novelty Model Development

6.1.1 Architecture Selection

The architecture selected is similar to the one presented in [62] and is a modified version of the variational Auto-Encoder (VAE) described in subsubsection A.1. The architecture was selected due to its high performances [71, 62]. As previously mentioned, in [62], the model was trained using subsets of known classes, leaving the rest of the classes to be used as unknown during testing. The model was capable of correctly classifying known classes while correctly identifying unknown classes since it learned to reconstruct known classes properly. Similarly, applying this type of model in an aviation context, a model is trained on known anomalies and nominal data such that the model can capture the current knowledge of flight analysts.

The encoding portion of the model in Figure 6.1 embeds the input into a lower dimensional latent space. The latent representation of the input then goes through a classifier, composed of a linear layer, that is used to classify the input. The encoding part, which receives flight data as input, contains 5 convolutional blocks, each block being an succession of 1-D CNN, batch normalization, a ReLU activation function, and a fully-connected layer to flatten the output as seen on Figure 6.2. The CNN channels in each block are gradually increased while the latent representation dimensions are reduced. The model hyperparameters are shown in Table 6.1. In Figure 6.2, h is the input to the CNN in the next convolutional block and μ and σ are the latent outputs for that given block. The latent space of the final convolutional block is then the input to the decoder, which reconstructs the original input.

Like the encoder, the decoder part is composed of transposed convolutional blocks, but as opposed to the former, the decoder's CNNs have gradually increasing channels and decreasing latent dimensions. Each decoder takes the corresponding encoder's latent representation, and the previous transpose convolution blocks latent representation as inputs

and outputs a latent representation of gradually higher dimension, as seen on Figure 6.1. In practice, the probabilistic ladder architecture enables the computation of q_μ and q_σ using the outputs μ and σ of the encoder and the outputs $\tilde{\mu}$ and $\tilde{\sigma}$ of the decoder as defined in [62] and in [71]. It is important to note that the last transpose convolutional block outputs a reconstruction of the input rather than computing another latent space. The transposed convolutional layers are composed of fully connected layers that convert the latent space vectors back into the appropriate tensor shape for a convolutional neural network (i.e. unflattens the input). It is then followed by a 1-D transposed convolutional neural network, a batch normalization layer, and a fully-connected that flattens the output. The decoder's hyperparameters are symmetrical to the encoder's ones, as it is typically done for Auto-Encoders. As mentioned in the literature review, Auto-Encoder are useful to detect abnormal patterns and have been successfully used in diverse applications.

This architecture is different from a standard VAE due to the probabilistic ladder used, and the changes made to the KL-divergence in the latent space, now given by:

$$KL(q_\phi(z|x)||p_\theta(z)) = \frac{1}{2}\sum_{j=1}^{J}(1 + \log(\sigma_j^2) - (\mu_j - \mu_j^{(k)})^2 - \sigma_j^2) \tag{6.1}$$

Where $\mu_j^{(k)}$ is the mean of the k-th Gaussian distribution, with k being the index of the known classes. Additionally, due to the probabilistic ladder architecture, the KL divergence is also defined in middle layers as explained in [62].

Table 6.1: Hyperparaneters of Novelty Model's Encoder

Convolutional Block Number	Kernel Size	Output Channels	Stride	Latent Dimensions
1	5	32	1	256
2	3	64	1	128
3	3	128	1	64
4	3	256	1	32
5	3	512	1	16

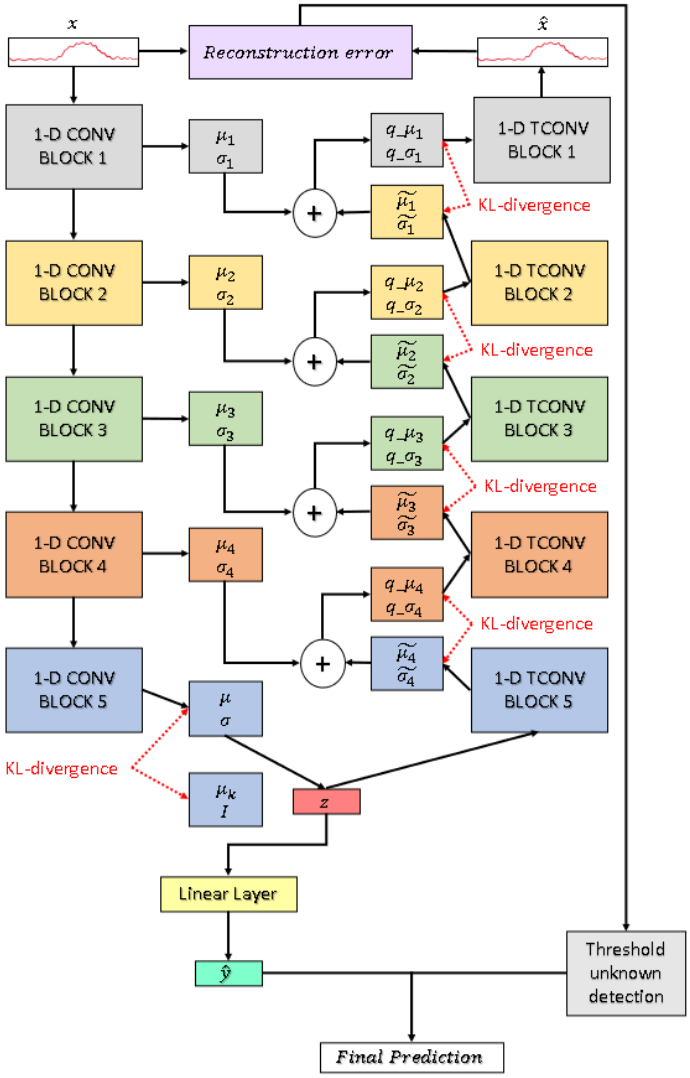


Figure 6.1: Modified Probabilistic Ladder Architecture

This model provides the capability to detect potentially abnormal behaviors and anomalies that analysts did not expect. Neural networks were chosen for the novelty detection task as the long-term goal is to develop a unique precursor model that has both novelty detection and precursor mining capabilities, therefore yielding an all in one model. Having such a model would help reduce the need for multiple models.

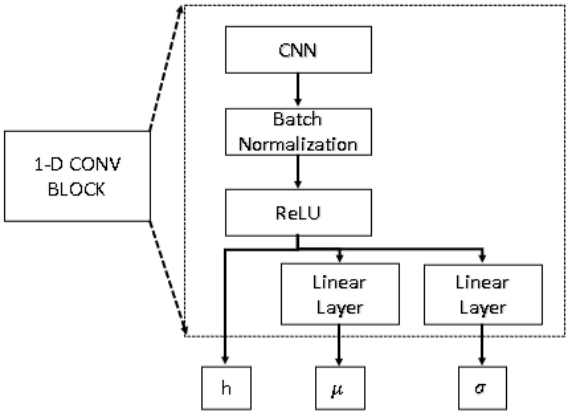


Figure 6.2: Composition of The Convolutional Block

6.1.2 Novelty Detection and Model Evaluation

After the model is trained, the reconstruction error R for each flight is computed by taking the sum squared error between each reconstructed feature and the original feature value as follow:

$$R = \sum_{i=1}^{D} (\hat{x_i} - x_i)^2 \tag{6.2}$$

Where D is the number of features, $\hat{x_i}$ is the reconstructed parameter i and x_i is the original input parameter i. Similar to [62], a threshold is set on the reconstruction error. For this work, the threshold was set such that 99% of the validation data is considered to be known, while the remaining would be flagged as unknown. The model is evaluated similarly to a binary model since the goal is to detect whether a new flight is either nominal or from the known list of anomalies, or whether the flight corresponds to a novel anomaly that was not know before. Thus metrics highlighted in subsubsection 4.4.2 such as the confusion matrix, and the F1 score were used to evaluate the model. When the novelty detection model and the precursor models are combined the Macro F1 score is used:

$$\text{Macro F1} = \frac{1}{C} \sum_{i=1}^{C} F1_i \tag{6.3}$$

Where C is the number of class and $F1_i$ is the F1 score obtained when considering each class (nominal, high-speed, high path angle/flaps late) positive at a time.

6.2 Experiment 3

6.2.1 Purpose of Experiment

This experiment aims to be a proof of concept and to demonstrate that:

1. Models can be trained such that they learn to represent nominal and pre-defined event classes, and therefore can flag new flights that do not belong to these classes
2. Combination of the novelty detection model and the precursor mining model can ensure better forecasting performances, notably when novel data is present

6.2.2 Experiment Setup

The data was pre-processed using the same steps described in chapter 4. Afterward, the model was trained using only nominal flights and flights that experienced a high-speed event. As previously mentioned, a threshold is set on the reconstruction error such that there are constant true negative (99%) and false positive (1%) rates on the validation data. That threshold is then used to classify new flights as known or unknown. The performances of the algorithm towards classifying known and unknown classes are evaluated using the F1 score and the confusion matrix. New flights belonging to the high path angle event and to the flap late event classes were used to test the model. It was reported in chapter 4, that the high-speed and the high path angle event had some common precursors. Therefore, late flap event is introduced in this section as a new event in order to test the performances of the novelty detection algorithm on an event that is not similar to the ones that were studied previously. The novelty model is then combined with the precursor model trained previously. A test data set composed of all classes (all events and nominal operations) is

then given as an input to the combination model. This model performance is evaluated by comparing its F1-macro to the one of the precursor model alone.

6.2.3 Experiment Results

6.2.4 High-Speed Known Event Classification

The novelty detection model was trained to recognize known classes. The model performances are therefore evaluated using common classification metrics. Table 6.2 summarizes the scores obtained on the test set before adding novelty to it. The scores showed that the classifier part of the architecture is able to learn from the data just like the precursor model. In fact, for the high-speed event, the models have similar performance but the novelty detection algorithm does not provide precursor identification capabilities. The model's latent space of dimension 16 is reduced to 2 using PCA for visualization as seen on Figure 6.3. It can be inferred from the figure that the model is able to have accurate predictions of the known class because it can separate them.

Table 6.2: Novelty Detection Algorithm High-Speed Classification Performances

F1 Score	**Precision**	**Recall**
0.91	0.89	0.93

Flaps Late Unknown Event Detection

After the threshold was set, the trained model was evaluated on a testing set. Table 6.3 provides the distribution of the reconstruction error on the validation set, which contains only nominal and high-speed event data. Selecting the threshold for the reconstruction error to be $2.67e^6$ ensures that most of the flights of the known classes will be correctly labeled as known. Flights that experienced a late flaps event were added to the set so that the model's ability to detect novelty could be evaluated. Results obtained demonstrated

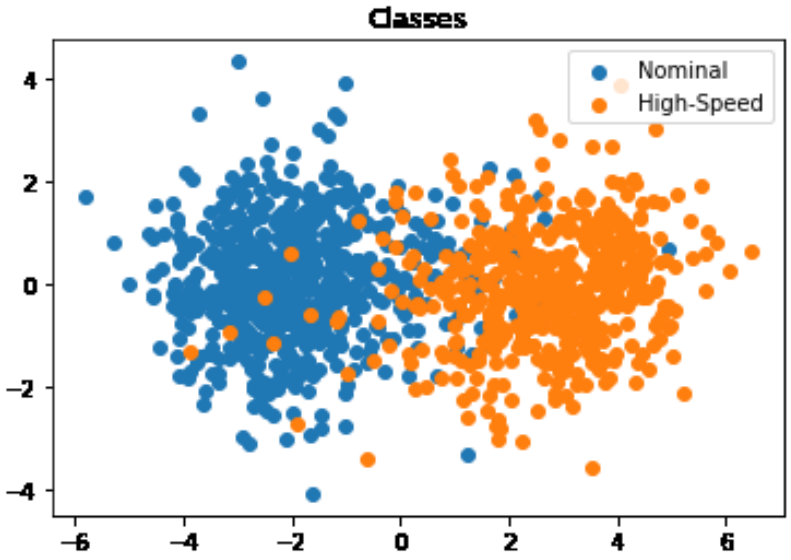


Figure 6.3: Lower Dimensional Representation of Novelty's Model Latent Space

Table 6.3: Validation Set Reconstruction Error Distribution

	Reconstruction Error
Mean	$4.43e^{5}$
Standard Deviation	$8.31e^{5}$
Minimum	2.95^{-1}
10%	$7.96e^{3}$
25%	$5.08e^{4}$
50%	$1.88e^{5}$
75%	$5.39e^{5}$
90%	$1.15e^{6}$
99%	$2.67e^{6}$
Maximum	$1.89e^{7}$

that the algorithm was capable of identifying the flaps late event as a novelty. Overall, the model's performances were satisfying for this event as seen on Table 6.4, although the model seems to be biased towards false positives since it incorrectly predicted 34 flights to unknown and 0 to be known. In addition to the model performances, the latent space can be visualized on a 2-D plot using PCA as seen on Figure 6.4 to get some insights into the model decisions. When introducing the flaps event, it can be seen that the latent space representation of flights that experienced the event is very different from the ones

of the known events. This difference in the latent space confirms that the model clearly differentiate between the known classes and the unknown one.

Table 6.4: Confusion Matrix (Novelty Detection-Flaps Late Event)

	Predicted Known	**Predicted Unknown**
Actual Known	3440	34
Actual Unknown	0	306

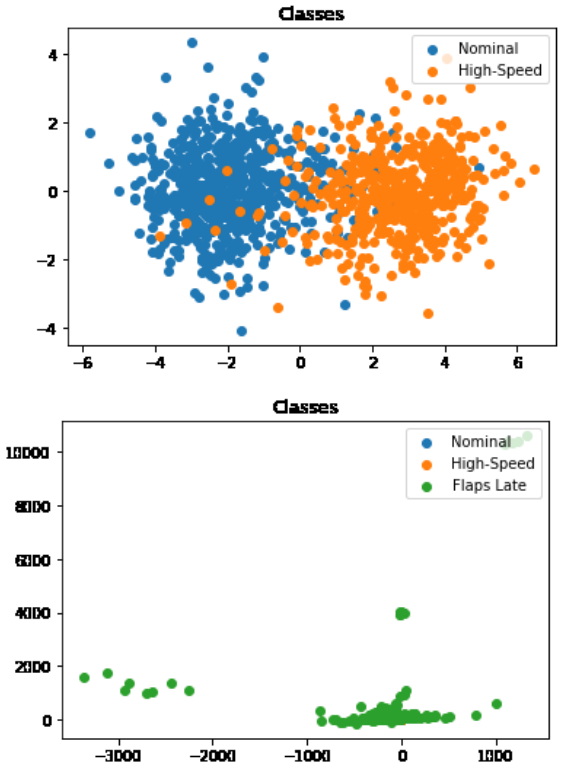


Figure 6.4: Nominal (Blue), High-Speed Event (Orange), and Flaps Late Event (Green)

High Path Angle Unknown Event Detection

A similar experiment was also run for the high path angle event. The novelty detection algorithm was given flights that experienced this event to mimic novelty, and the behavior of the model was observed. Figure 6.5 shows the 2-D representation of the 16 dimensional

latent space. Like Figure 6.4, the flights that experienced the high path angle event do not overlap with the known classes. This visualization suggests that the model is again capable of identifying novelty in the data even though more novel data points were available and used. It is indeed confirmed by the results obtained, as shown by the confusion matrix in Table 6.5.

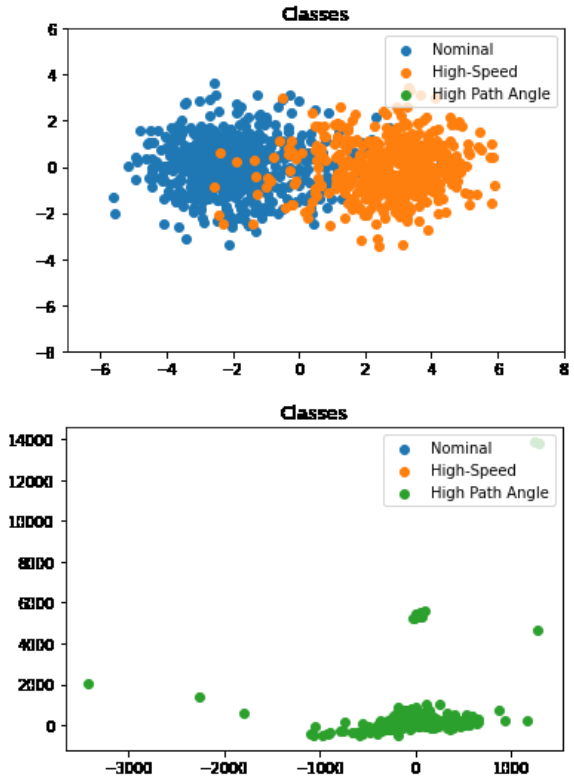


Figure 6.5: Nominal (Blue), High Speed Event (Orange), and High Path Angle Event (Green)

Table 6.5: Confusion Matrix (Novelty Detection-High Path Angle Event)

	Predicted Known	**Predicted Unknown**
Actual Known	3435	39
Actual Unknown	0	1089

6.2.5 Enhancing The Precursor Model for Real-World Settings

The main goal of training the novelty detection algorithm is to ensure that the precursor model is used within its limits. In a real-life setting, the precursor model might see new events that were not encountered during the training and perform worse than expected. Therefore, the novelty algorithm is combined with the precursor model to enhance the latter's performances when new events are introduced. Figure 6.6 details the framework to combine both models. Flight data belonging to either nominal, high speed (anomaly 1), and either high path or flaps late (anomaly 2) goes through the trained Auto-Encoder first. This model decides whether the input is known or unknown. If the input is known, it is then sent to the previously trained precursor model that classifies it as nominal or as a high-speed event. The performances of these combinations are evaluated by comparing the F1 macro scores before and after the introduction of the Auto-Encoder.

Table 6.6: Combined Model Evaluation Results (Novelty with Flaps Late Event)

Event	Model	Macro Average F1 Score
Flaps Late	Precursor Model	0.53
Flaps Late	Novelty Model + Precursor Model	0.86
High Path Angle	Precursor Model	0.49
High Path Angle	Novelty Model + Precursor Model	0.88

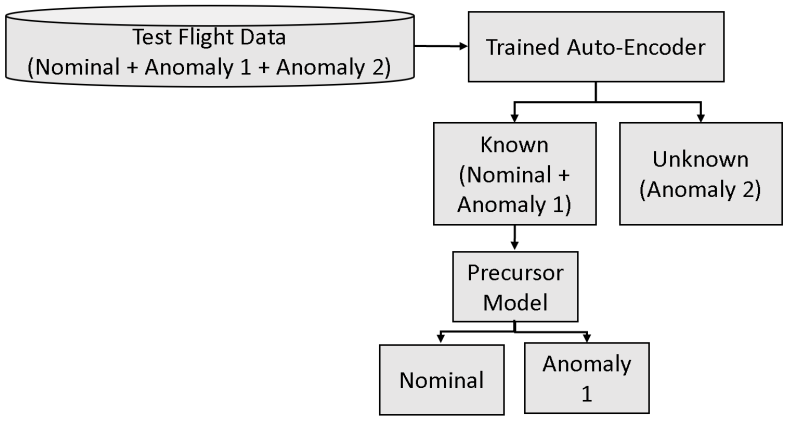


Figure 6.6: Framework for Combining Novelty and Precursor Models

6.2.6 Discussion

Overall the results of the experiment are encouraging. The novelty detection model is capable of correctly classifying nominal and high-speed events, which were known classes. When introducing a new event such as the flaps late and the high path angle, the model is able to detect novelty though it is biased towards false positives. The benefits of adding the novelty detection model were observed when the novelty detection model was combined to the precursor model resulting in better performances when novel data is present. The Auto-Encoder essentially behaves as a filter and enables the precursor model to meet its closed-set assumption. The F1 scores of cases that used the combined models were significantly better than those that only used the precursor model for each event. In addition to the classical classification metrics, the latent space's visualization was also valuable to understand the model's behavior. The latent space visualization showed that the model is able to create latent normal distributions for each of its training classes, and that these distributions are separable. The latent space representation of the novel events (high path angle and flaps late) were significantly different from the ones of the nominal and high-speed event.. Finally, the results provides enough ground to support **hypothesis 3**.

CHAPTER 7
CONCLUSION

The aviation industry brings tremendous social and economic benefits to the world. The industry has been rapidly growing over the past decades leading to milestones in the world-wide number of passengers. A crucial factor of the growth is safety. Indeed safety is essential for aviation as its lack could result in death and extensive damages to property. Before COVID-19, it was expected for the industry to keep growing. With this expected growth comes more passengers, more complex operations, and more risks. Therefore, the industry must reduce safety rates as they have been stagnating in recent years. The constant safety rates are likely due to the limitations of some of the currently implemented measures. Aerospace systems are becoming more complex, and they generate an increasing amount of data, which makes getting insights from this data and understanding the causes of incidents and accidents difficult. Current techniques are manual and not scalable. Besides, lots of existing systems are reactive, leaving pilots with insufficient time to respond to imminent danger.

This thesis proposed exploring a novel precursor mining method to improve aviation safety: The Intelligent Methodology for the Discovery of Precursors of Adverse Event (IM-DoPE). Precursor mining is expected to enhance safety performances since it provides predictive capabilities and can forecast events before they occur and understand why the events happened. This research highlights several gaps in current modern precursor mining algorithms. It builds on top of them by combining the feature extraction capabilities of Convolutional Neural Networks (CNNs) with Recurrent Neural Networks (RNNs) to process time-series and retrieve the input aircraft parameters importance. This work successfully developed models capable of forecasting known adverse events such as high-speed and high path angle events and identifying their precursors. Next, a standard methodology

levering precursor scores (common metric to modern precursor mining technique) was implemented. The methodology was used to cluster flights obtain granular insights into the possible cause of the studied adverse events. Finally, this work proposed a way to ensure that the created precursor models are used within their limits, and that novelty is detected by combining the developed precursor model with the novelty detection model.

7.1 Review of Research Questions and Hypotheses

7.1.1 Research Questions

A literature review was performed, and the several gaps in current methodologies were highlighted in subsection 2.3.4. These gaps led to the formulation of the following research questions:

- **Research Question 1:**How can current temporal-based data mining methods be improved to identify precursors of adverse events and account for their potential interactions?
- **Research Question 2:** How can a standardized approach take advantage of identified precursors to discover potential causes of multiple adverse events?
- **Research Question 3:** How can the lack of data be compensated for so that the created model's usefulness is ensured?

In particular, the incapacity of retrieving the input's contributions when using neural networks and the need for bias sensitivity analysis to cope with that motivated research question 1. Research question 2 was motivated by the lack of a common methodology capable of leveraging any precursor algorithm's precursor scores and using them to discover potential causes of an adverse event. Additionally, none of the current methods explicitly addressed how to find multiple causes for the same event. Finally, research question 3 was formulated to address the lack of precursor mining techniques utilizing unsupervised learn-

ing and free models from the common closed-set assumption used by all the supervised precursor mining methodologies.

7.1.2 Hypotheses

Three hypothesis were associated with these research questions. Research question 1 is mapped to the first hypothesis:

Hypothesis 1: *If deep learning methods that extract temporal information are extended such that individual and combined contributions of aircraft parameters are automatically retrieved, then precursors will be identified without any bias.*

The literature review identified the Automatic Discovery of Precursors in Time Series Data (ADOPT) as the most promising precursor mining method. The model achieved high performances and inherently could process sequences thanks to the RNNs used, unlike other precursor mining methods. A significant flaw of the model is that it requires a post-processing sensitivity analysis to identify precursors of studied events because the contributions of each input parameters and their combination are lost due to the Recurrent Neural Networks used by the algorithm. Adding a Convolutional Neural Network (CNN) to the Recurrent Neural Network (RNN), therefore, extended the model. In particular, the CNNs could retrieve the precursors influencing the adverse event without the required sensitivity analysis. Experiment 1 was setup to use the created model to retrieve precursors to high-speed and high path angle events, and confirm the abnormal behaviors exhibited by the identified precursors. **Hypothesis 1** was validated by visualizing and observing significant differences in the original flight data of nominal flights and flights that experienced either of the adverse events studied for the identified precursors.

Research question 2 is associated with hypothesis 2:

Hypothesis 2: *If precursor scores are computed for the identified precursors and used*

to cluster flights that experienced adverse events, then the clusters will be analyzed to discover potential causes of these events.

In the literature, precursor scores were computed in some of the precursor mining methods, and therefore the proposed methodology defines a standardized way to use them. The inputs to this part of the methodology are the precursor scores, which can be extracted from any mining method. Additionally, clustering flights using their precursor scores for an event of interest means that flights with similar precursors are grouped and exhibit similar abnormal behaviors. Therefore the methodology identifies multiple potential causes for the same event. Experiment 2 aimed to demonstrate that flights can be grouped using their precursor scores, and that the created groups provide different insights into a single event. **Hypothesis 2** was validated as the visualizations of the original flight data comparing nominal to abnormal flights highlighted important differences in the identified precursors. Experiment 2 showed that different abnormal behaviors were be observed from one cluster to another, though the adverse flights within these clusters all experienced the same adverse event.

The final research question 3 mapped to hypothesis 3:

Hypothesis 3: *If an anomaly/novelty detection algorithm is used to flag new anomalies that were not pre-defined and combined with predictive models that learned to recognize defined anomalies, then the created predictive models will be used within their limits.*

The machine learning and deep learning models used in the literature and the model developed for this thesis all receive an incomplete knowledge of the world during training. Indeed, the models learn from the data available in the training set, but they might encounter unknown data that was not present in the training data when used in the real world. Experiment 3 was run to first show that a model can be created to learn to represent the current knowledge of analysts (nominal conditions and known anomalies). The second goal of

experiment 3 was to demonstrate that combining a model that captures current knowledge with a precursor model, allows the latter model to perform better in the real-world where data is not limited to the events present in the training data. **Hypothesis 3** was validated as the results of the experiment showed that the created model was able to detect novelty in the data it received, and that pairing this model with the precursor model enabled better performances when novelty was introduced.

The validation of the three hypotheses made it possible to answer the research questions of this thesis and overall achieve the research objective that was set.

7.2 Benefits of This Work

Multiple benefits are expected from this work. First, stakeholders such as airlines, the National Transport Safety Board (NTSB), aircraft manufacturers, and even start-ups focusing on the Urban Air Mobility field are expected to find this work useful. This novel precursor mining method helps to improve predictive capabilities with models that can help prevent an incoming adverse event. Deploying this work online would provide real-time help to pilots as they fly the aircraft.

When used offline, this work can still help stakeholders automatically identify potential causes to pre-defined events using their precursors. The proposed methodology enables fast identification of the parameters that led to the event. Therefore it saves time for flight analysts who can directly focus on the flagged parameters. Also, the proposed usage of the precursor scores to cluster flights would allow the flight analysts to detect multiple possible starting points to understand the adverse event.

For academics, the models developed are novel for the aerospace industry and provides an additional capability to tackle precursor mining. This work offers improvements to previous methods and could be used for future benchmarking exercises. The proposed methodology also brings a common approach that leverages precursor scores of a given precursor mining algorithm to provide a more granular explanation of a studied adverse

event. The methodology is algorithm agnostics and only requires scores to be generated from the precursor mining method, as seen in [17] and [20].

7.3 Future Work

7.3.1 Model Interpretability

Although this research provided additional explainability when compared to ADOPT, the results obtained are still empirical and the interpretability of the model is still limited. A logical extension of this work would be to modify the loss function used to train the model in this work. The function should penalize the output the CNN such that is resembles the output the RNN, allowing for a better interpretation of the precursor score of each parameter over time. Additionally because not all parameters are important, the function should penalize the model such that no more than a certain number of feature maps obtained from the CNN should be matching the output of the RNN. Additional efforts could also include the use of SHAP (SHapley Additive exPlanations) [72] to provide additional interpretation capabilities to the neural network, which could help better understand the causes of events.

7.3.2 Identification of Precursors of Unknown Events

Promising results were obtained when the novelty detection model was trained on nominal and high speed flights and the novelty was introduced. Though the algorithm could be refined to obtain even better results, it would be more interesting to develop a one-in-all model that combines the novelty detection and the precursor mining capabilities. Multiple deployed models can be hard to manage, and having only one model would reduce unnecessary deployment efforts. More efforts should focus on understand the detected unknown classes. Flights analysts would benefit further from this work if the model latent space representations of the high path angle and the flaps late events showed new distinguishable clusters. This would help analysts understand how similar the flights that experienced unknown events are. Finally, similar to the work proposed in [22] the models developed for

this thesis could be used to identify precursors of unknown events.

7.3.3 Extension to Other Events

The algorithm was extensively tested on the high path angle event and the high-speed event as an hyperparameter search was completed for each of them. The search enabled the development of high-performing precursor models. It would be therefore interesting to investigate additional events to evaluate the performance of the methodology for other events. Moreover, the precursor model is limited by the amount of labeled since the model performance decreased for the high path angle event, which had a lower amount of available flights. It is then of interest to investigate the minimum required number of flights to develop a good model.

Appendices

APPENDIX A
MACHINE LEARNING AND DEEP LEARNING ALGORITHMS

A.0.1 Activation Functions

In deep learning, individual neurons perform a weighted sum of some inputs. The inputs can be coming from the original input data or from the output from a previous layer. The activation function is a function that takes as input the weighted sum, and decide if a given neuron can be fired or not [73]. They are useful to convert linear outputs of neural network to non-linear ones , which is especially useful to learn patterns in data. Different activation functions will lead to different results in deep networks, and could be thought of as parameters to include in a hyperparameter search. Furthermore, the required output of the neural network determines the final activation function that should be used. For instance, it is common to use sigmoid, softmax, or linear activation functions for binary classification, multi-class classification, and regression tasks respectively. The activation functions used for this work are described in this Appendix.

Sigmoid Activation Function

The sigmoid activation function is a bounded differential real function usually used with feedforward neural networks [73]. The inputs to the function are also real values. The function outputs are always positive and bounded between 0 and 1, making them useful when outputting probabilities. The function does have drawbacks, which include:

- Sharp damp gradients during backpropagation, especially in deep neural network
- Gradient saturation, for high positive and low negative inputs
- Slow convergence and non-zero centered output, which can cause issues in the gradient updates

Finally, the sigmoid function is defined by:

$$\sigma(x) = \frac{1}{1 + e^x} \tag{A.1}$$

Rectified Linear Unit (ReLU) Activation Function

The ReLU function is by far the most widely used action function in deep learning [73]. Its popularity comes from the advantages that it provides over other activation functions. These advantages are the faster computations, the preservation of properties of linear models such as the ease to optimize them via gradient descent, eliminating the vanishing gradient problem, better performance of deep learning models. The ReLU function is defined mathematically by:

$$f(x) = \max(0, x) = \begin{cases} x_i, & \text{if } x_i \geq 0 \\ 0, & \text{if } x_i < 0 \end{cases} \tag{A.2}$$

On the other hand, drawbacks of the ReLU function include: higher likelihood of overfitting, can cause of the gradient to die causing neurons to be "dead" or inactive, which can hinder the learning process.

A.1 Algorithms

K-means

The K-means algorithm can create k different clusters from a given data set, when given k as an input. The data set is usually in a table format with multiple features provided for many data points. The algorithm can be divided into the following steps [70, 74]:

1. Initialize k cluster centers c_1, c_2, ..., c_k randomly or by using a point in the data set

2. Assign the nearest cluster for each point $x_i \in \mathbf{R}^n$ according to a distance metric d

$$\pi(i) = \underset{j=1,\ldots,k}{\operatorname{argmin}} d(x_i, c_j) \tag{A.3}$$

3. Adjust the cluster center such that:

$$c_j = \underset{v \in \mathbf{R}^n}{\operatorname{argmin}} \sum_{i:\pi(i)=j} d(x_i, v) \tag{A.4}$$

4. Repeat 2,3 until no cluster center has changed

Deep Feedforward Networks

Artificial Neural Networks (ANN) are usually composed of one input layer, one hidden layer, and one output layer. Deep feedforward networks extend ANNs by creating a more complex topology with more than one hidden layer. Other names for deep feedforward networks are feedforward neural networks, and Multi-Layer Perceptrons (MLP). Each layer can contain multiple neurons, which are the basic computing units that receives its inputs from the input data, or from a previous layer. Each processing unit perform a weighted sum between its inputs and learnable parameters **w** as seen in Equation A.5.

$$f(x) = a(\mathbf{w}^T\mathbf{x} + b) \tag{A.5}$$

Where $a(\cdot)$ is an activation function, **w** and b, the weights and bias respectively are both learned parameters. Through a learning algorithm called backpropagation involving gradient descent, the neural network defined by the function f with $\mathbf{y} = f(\mathbf{x})$ attempts to approximate a true function f^* by learning its parameters. The parameters are updated after each backpropagation to minimize the error between the network output and the target. An example of a feedforward neural network is depicted in Figure A.1.

The structure is typically fully connected such that a single neuron inputs are the out-

puts from all the neurons of the previous layer. This type of neural network is the backbone of deep learning architectures as more complex ones such Convolutional Neural Networks (CNN) or Recurrent Neural Networks (RNN) are special cases of this one. With this architecture, an input $\mathbf{x}$ gets propagated in a feedforward manner (no feedback connection) through multiple layers to find an output $\mathbf{y} = f^{(3)}(f^{(2)}(f^{(1)}(\mathbf{x})))$ [75], where $f^{(i)}$ is the i^{th} layer, and the number of layers represent the depth of the network.

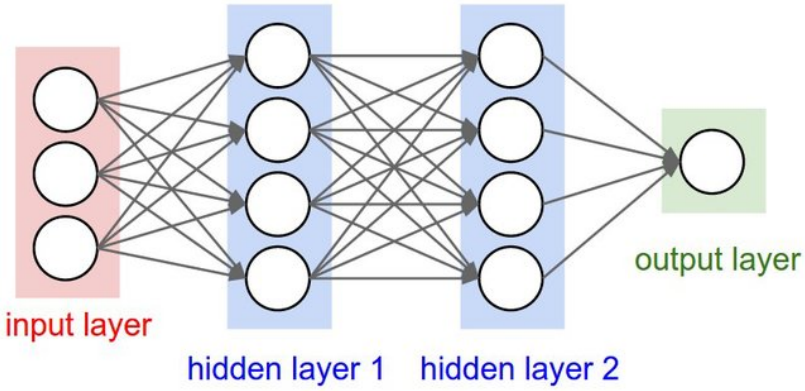


Figure A.1: Deep Feedforward Neural Network

Neural networks have multiple advantages such their ability to estimate non-linear functions, parallel processing capabilities, fault tolerance [76]. Disadvantages come from the difficulty in interpreting them, the hardware dependence that can be needed to train them, the amount of data required to train them [76].

Convolutional Neural Network

Convolutional Neural Networks (CNNs) can be thought of as analogous to traditional Artificial Neural Networks because they also have neurons with parameters that can be learned. They have been uses for diverse applications such as image classifications with 2-D and 3-D CNNs, and more recently time series with 1-D CNNs. The structure of 2-D CNNs is different because its layers are organized into three dimensions, which are the height, and the width both related to the input's spatial dimension, and the depth [77]. The height and width are usually defined by a kernel of a set size, and the depth can be thought of as the

number of dimensions when working with multivariate time-series or as the RGB layers of an image. Figure A.2 shows a kernel of size W_L slide over a time-series for a 1-D CNN. For these CNNs, only the height and the depth are used. The name of this type of neural network come from the mathematical operation used, a convolution [75] though the definition of convolution in deep learning does not exactly match the one used in mathematics or engineering. The following equation defines a discrete convolution operation along one dimension:

$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)w(t-a) \tag{A.6}$$

Where x is the input, w represents the learnable weights of the kernel, and s the output referred to as a feature map.

Additionally, in practice the sum is finite. A simplified 1-D convolution is depicted in Figure A.3 where the kernel size is 2, the input size is 5, and the feature map size is 4. As

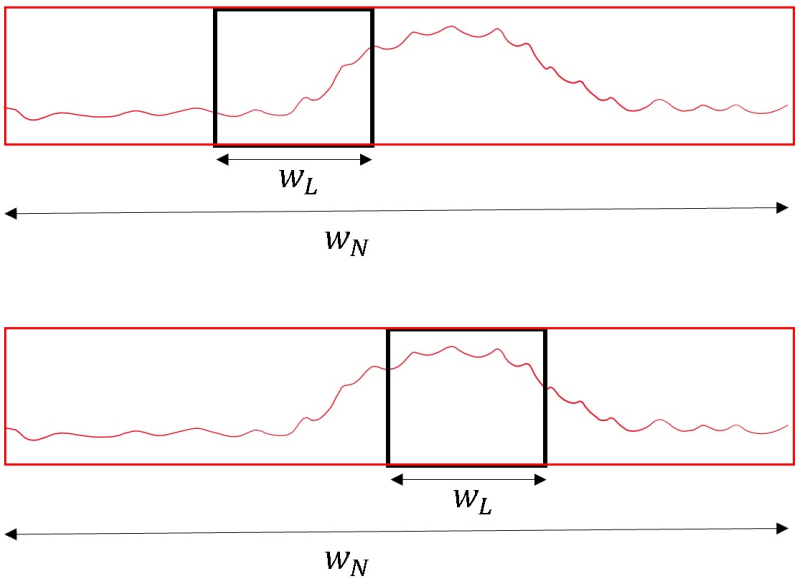


Figure A.2: Kernel of Size W_L and Stride =1 Sliding Over Time Series of Length W_N

seen on the figure, the weights of the kernel are applied at different regions of the time-series, and a linear operation, similar to Equation A.5, is performed to fill in every cell of the feature map. The notion of stride can be understood from the figure, where the stride

is the displacement size made by the kernel before computing the next cell of the feature map. The stride in this example is of 1. The feature map size can be predetermined as it is a function of the size of input, the size of the kernel, the stride, and whether padding was used as by the following equation:

$$n_{out} = \frac{n_i n - 2p - f}{s} + 1 \tag{A.7}$$

With $n_i n$ being the spatial length on the input, p the padding size, kernel (or filter) size, s the stride, and n_{out} is the length of the feature map. In the case where no padding is used, the feature map size is smaller than the input size, yielding a "valid" type of convolution. Padding can be used to add additional border. For example, padding the presented time-series in Figure A.3 would result in a longer input: $[0, 2, 3, 4, 1, 6]$ which would yield a feature map of size similar to the original input without any padding, if the same kernel of size 2 is used. This case would result in a type of convolution named "same." A common value used to pad is zero. Moreover, the example shown assumes that only one filter is

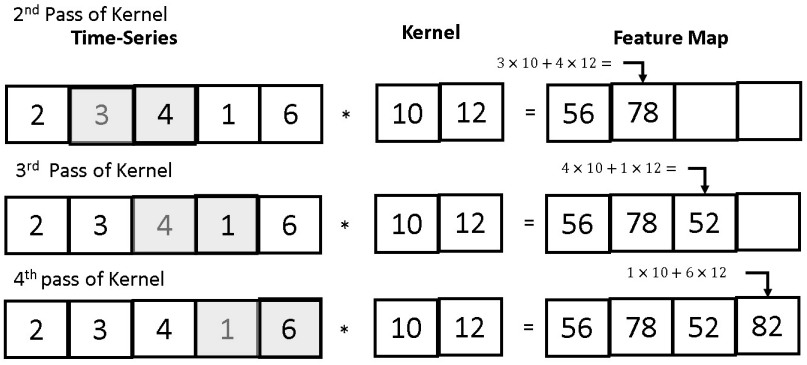


Figure A.3: Kernel Passes Over Example Time-Series

applied resulting in a feature map of channel/depth 1. If n more filters were applied, then the final feature map would have have a depth of n. Finally in practice, a bias a is added during the convolution and the feature map created is passed through an activation function.

Gated Recurrent Unit

Recurrent Neural Networks (RNNs) are another type of neural networks particularly suited to handle sequential data of values $x^{(1)}, \ldots, x^{(\tau)}$ [75]. Feedforward Networks cannot model sequences since they have fixed sized inputs and outputs, and no way of capturing the temporal structure. Additionally they have no memory, and no feedback since they only process the data in a feedforward manner. A notion of shared parameters is present in RNNs. The shared parameters across different time-steps enables the generalization of the model because it can be applied to examples of different lengths, the parameters are not fixed to the time index. It also enables the sharing of statistical strength across different time instances, which can be important since relevant information can be anywhere is the sequence. Common applications of RNNs include time-series classification, speech-to-text, handwritten notes to text, machine translation, and more,

Gated Recurrent Units (GRUs) are a particular type of RNNs. They were introduced as an alternative to the more complex Long Short Term Memory (LSTMs) neural networks[17]. The GRU is simpler because it has fewer number of gates which are: the reset and update gates. Both of these networks were developed to handle the short-term memory problem that other RNNs have. The internal structure of GRUs is presented in Figure A.4, and its mathematical formulation is as follow:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \tag{A.8}$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \tag{A.9}$$

$$\tilde{h}_t = \tanh(W \cdot [r_t * h_{t-1}, x_t]) \tag{A.10}$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t \tag{A.11}$$

Where W_r, W_z, and W are the GRU parameters that are learned through the learning algorithm. Other variables x_t, z_t, and r_t are the input, the update gate, and the reset gate. Furthermore, h_t is the hidden state and $\tilde{h}_t$ is the intermediate memory unit. The reset gate is responsible for deciding whether the previous hidden state or the input at time t should be ignored or not. The output is bounded, due to the sigmoid function, between 0 and 1, where 1 allows the information and 0 doesn't. The update gate is responsible for deciding if new information gets added to the hidden state. From Equation A.8, the closer to 1 z_t is the less impact the input x_t will have on updating h_t. The intermediate $\tilde{h}_t$ also plays a role in update h_t. As the GRU is "unrolled" through time, multiple units with each having inputs $x^{(1)}, \ldots, x^{(\tau)}$ are obtained, and the network is therefore allowed to learn the temporal pattern within the data.

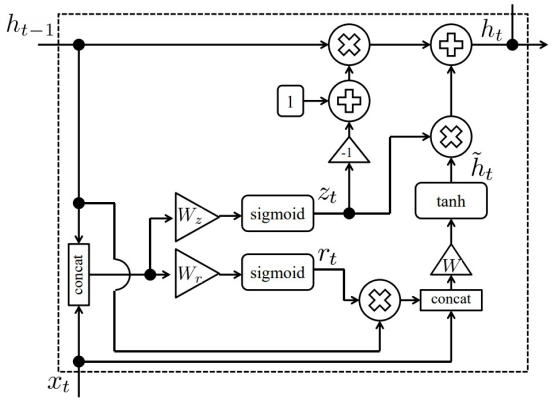


Figure A.4: Internal Structure of a GRU Unit

Variational Auto-Encoder (VAE)

Auto-Encoder are neural networks that learns to output the reconstruction of an original input. They are usually used for unsupervised learning tasks, which enables learning a lower-dimensional feature representation from unlabeled training data. The network usu-

ally gradually reduces the dimension of an input up to a bottle-neck, which correspond to a compressed representation of the original input. The part of the network responsible for the dimensionality reduction is the encoder. Encoding is required to recognize significant factors of variation in the data. The bottle-neck is the input to the decoder, which gradually increases the dimensionality back to the original input's dimension. The loss function usually measures how well the reconstructed input resembles the original one.

The Variational Auto-Encoder (VAE) are a particular type of neural networks that adds a probabilistic spin to the regular Auto-Encoder. In particular, the latent space of the VAE is described using a probability distribution making the model a generative one. The model is composed of an encoder with parameters ϕ and a decoder with parameters θ. Mathematically, the VAE is described as such: for a given input x, to estimate the conditional density of the posterior distribution $p(z|x)$, where z is the latent variable [78]. Hence, the following equation :

$$p(z|x) = \frac{p(z,x)}{p(x)} \tag{A.12}$$

Unfortunately, computed $p(x)$ is a hard task, and usually turn out to be a intractable solution. Therefore, an attempt to perform an exact computation of a simpler distribution $q(z|x)$ that is close to the original complex one $p(z|x)$ is made. The Kullback–Leibler (KL) divergence is then used to measure the resemblances between the two distributions. In fact, the aim is to minimize KL:

$$\min KL(q(z|x)||p(z|x)) \tag{A.13}$$

which is equivalent to maximizing the evidence lower bound (ELBO):

$$ELBO = \mathrm{E}_{q(z|x)} \log p(x|z) - KL(q(z|x)||p(z)) \tag{A.14}$$

Where the first term correspond to the reconstruction likelihood, and the second makes sure

that the learned distribution is similar to the prior $p(z)$.

In particular, given the parameters for encoder, and decoder the loss function of the VAE for an input x is given by:

$$\mathcal{L}(x;\phi;\theta) = \mathrm{E}_{q_\phi(z|x)} \log p_\theta(x|z) - KL(q_\phi(z|x)||p_\theta(z)) \tag{A.15}$$

The prior $p_\theta(z)$ is usually taken to be the centered isotropic multivariate Gaussian [62], so that $p_\theta(z) = \mathcal{N}(z; \mathbf{0}, \mathbf{I})$. Additionally, the approximate distribution q_ϕ is defined by:

$$q_\phi(z|x) = \mathcal{N}(z; \mu, \sigma^{\mathbf{2}}\mathbf{I}) \tag{A.16}$$

Therefore, the KL-divergence is expressed as:

$$KL(q_\phi(z|x)||p_\theta(z)) = \frac{1}{2}\sum_{j=1}^{J}(1 + \log(\sigma_j^2) - \mu_j^2 - \sigma_j^2) \tag{A.17}$$

Where j is the index for the dimension of z.

APPENDIX B

ADDITIONAL FIGURES

B.1 Model Architectures

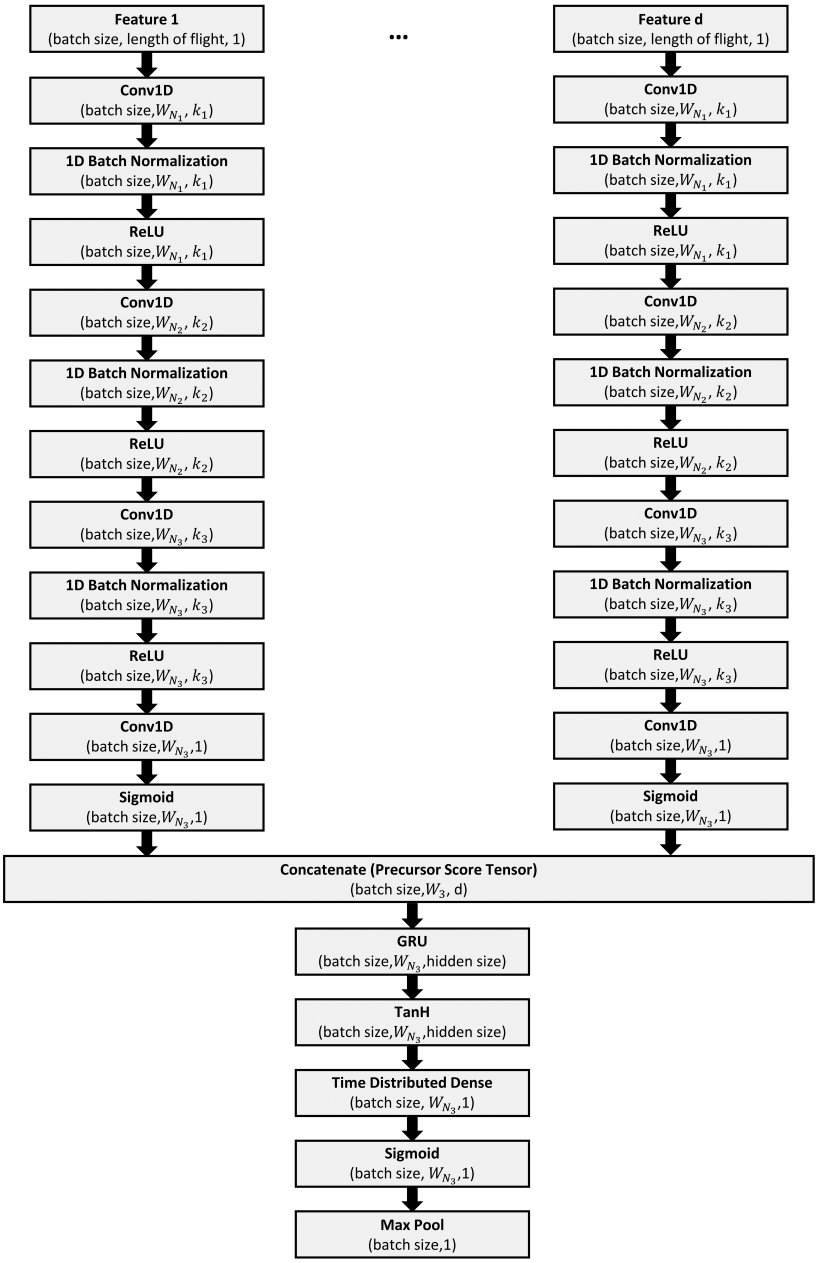


Figure B.1: IM-DoPE Precursor Model Architecture

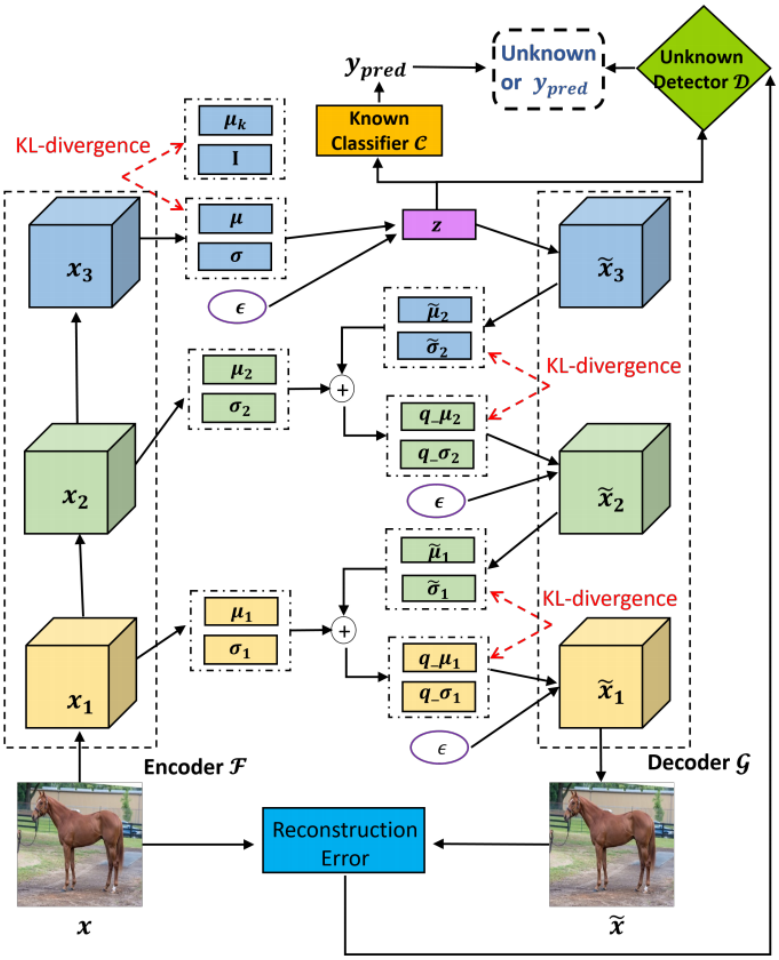


Figure B.2: Probabilistic Ladder Architecture [62]

REFERENCES

[1] ICAO, *Aviation benefits report*, https://www.icao.int/sustainability/Documents/AVIATION-BENEFITS-2019-web.pdf, 2019.

[2] N. R. Council, *An Assessment of NASA's National Aviation Operations Monitoring Service*. Washington, DC: The National Academies Press, 2009, ISBN: 978-0-309-14646-3.

[3] R. Allmond, *Aviation safety 2019 year in review*, https://www.faa.gov/about/office_org/headquarters_offices/avs/media/2019_Year_In_Review.pdf, 2019.

[4] IATA, https://libraryonline.erau.edu/online-full-text/iata-safety-reports/IATA-Safety-Report-2019.pdf, Apr. 2020.

[5] FAA, *Runway excursions*, https://www.faa.gov/airports/runway_safety/excursion/, journal=Runway Excursions, Aug. 2020.

[6] Federal Aviation Administration, *Airplane flying handbook*, https://www.faa.gov/regulations_policies/handbooks_manuals/aviation/airplane_handbook/media/10_afh_ch8.pdf, 2016.

[7] P. K. Menon, P. Dutta, O. Chen, H. Iyer, and B.-J. Yang, "A modeling environment for assessing aviation safety," in *AIAA Aviation 2019 Forum*. eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2019-2937.

[8] C. V. Oster, J. S. Strong, and C. K. Zorn, "Analyzing aviation safety: Problems, challenges, opportunities," *Research in Transportation Economics*, vol. 43, no. 1, pp. 148–164, 2013, The Economics of Transportation Safety.

[9] NASA, *ASRS - Aviation Safety Reporting System - Program Briefing*, https://asrs.arc.nasa.gov/overview/summary.html.

[10] B. Dunbar, *NASA Aviation Safety Reporting System Turns 30*, https://www.nasa.gov/home/hqnews/2006/nov/HQ_06345_ASRS_turns_30.html, Nov. 2006.

[11] T. Matsumura, C. Park, D. Doyon, R. Haftka, and N.-H. Kim, "Modeling the contribution of accident investigation to aircraft safety," in *10th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference*. eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2010-9057.

[12] FAA, *Federal Aviation Administration Advisory Circular, 120-82 - Flight Operational Quality Assurance*, url: https://www.faa.gov/regulations_policies/advisory_

circulars/index.cfm/go/document.information/documentID/23227, Federal Aviation Administration, Apr. 2004.

[13] THE BOEING COMPANY, *Aviation safety*, https://www.boeing.com/company/about-bca/aviation-safety.page.

[14] D. Gates, *Faa cautions airlines on maintenance of sensors that were key to 737 max crashes*, https://www.seattletimes.com/business/boeing-aerospace/faa-cautions-airlines-on-maintenance-of-sensors-that-were-key-to-737-max-crashes/, Aug. 2019.

[15] V. M. Janakiraman, "Explaining aviation safety incidents using deep temporal multiple instance learning," pp. 406–415, Jul. 2018.

[16] O. Wyman, *The data science revolution that's transforming aviation*, https://www.forbes.com/sites/oliverwyman/2017/06/16/the-data-science-revolution-transforming-aviation/, Jun. 2017.

[17] V. M. Janakiraman, B. Matthews, and N. Oza, "Using adopt algorithm and operational data to discover precursors to aviation adverse events," 2018.

[18] A. Gavrilovski, H. Jimenez, D. Mavris, A. H. Rao, K. Marais, S. Shin, and I. Hwang, "Challenges and opportunities in flight data mining: A review of the state of the art," 2016.

[19] L. Basora, X. Olive, and T. Dubot, "Recent Advances in Anomaly Detection Methods Applied to Aviation," *Aerospace*, vol. 6(11), p. 117, 2019.

[20] J. Ackley, T. G. Puranik, and D. N. Mavris, "A supervised learning approach for safety event precursor identification in commercial aviation," in *AIAA Aviation Forum*, 2020.

[21] H. Lee, H. J. Lim, P. Parker, and A. Chattopadhyay, "Precursor detection of aircraft loss of control in-flight (loc-i) and prediction of future trajectory," in *AIAA AVIATION 2020 FORUM*. eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2020-2879.

[22] R. Deshmukh, D. Sun, K. Kim, and I. Hwang, "Reactive temporal logic-based precursor detection algorithm for terminal airspace operations," *Journal of Air Transportation*, vol. 0, no. 0, pp. 1–9, 0. eprint: https://doi.org/10.2514/1.D0182.

[23] FAA, *Preliminary aviation statistics*, https://www.ntsb.gov/investigations/data/Pages/aviation_stats.aspx, 2018.

[24] J. G. Busquets, A. Evans, and E. Alonso, "Application of data mining in air traffic forecasting," in *15th AIAA Aviation Technology, Integration, and Operations Conference*. eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2015-2732.

[25] K. Sheridan, T. Puranik, E. Mangortey, O. Pinon, M. Kirby, and D. Mavris, "An application of dbscan clustering for flight anomaly detection during the approach phase," in *AIAA SciTech Forum*, Jan. 2020.

[26] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, Jul. 2009.

[27] N. Gugulothu, V. TV, P. Malhotra, L. Vig, P. Agarwal, and G. Shroff, *Predicting remaining useful life using time series embeddings based on recurrent neural networks*, 2017. arXiv: 1709.01073 [cs.LG].

[28] T. Puranik, "A methodology for quantitative data-drive safety assessment for general aviation," PhD thesis, Georgia Institute of Technology, 2018.

[29] E. Mangortey, D. Monteiro, J. Ackley, Z. Gao, T. G. Puranik, M. Kirby, O. J. Pinon-Fischer, and D. N. Mavris, "Application of machine learning techniques to parameter selection for flight risk identification," in *AIAA Scitech 2020 Forum*. eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2020-1850.

[30] Intel, *Advanced data analytics: Making your business smarter*, https://www.intel.com/content/www/us/en/analytics/advanced-data-analytics.html.

[31] McKinsey Global Institute, *The age of analytics: Competing in a data-driven world*, =http://tiny.cc/7xiysz, Dec. 2016.

[32] K. Murphy, *Machine Learning: a Probabilistic Perspective*. 2013, ISBN: 978-0262018029.

[33] C. Arthur, *Tech giants may be huge, but nothing matches big data*, https://www.theguardian.com/technology/2013/aug/23/tech-giants-data, Aug. 2013.

[34] SAS, *Big data: What it is and why it matters*, https://www.sas.com/en_us/insights/big-data/what-is-big-data.html.

[35] T. M. Mitchell, *Machine learning*. McGraw Hill, 2017.

[36] B. Marr, *27 incredible examples of ai and machine learning in practice*, https://www.forbes.com/sites/bernardmarr/2018/04/30/27-incredible-examples-of-ai-and-machine-learning-in-practice/, Dec. 2018.

[37] S. Minaee, *20 popular machine learning metrics. part 1: Classification amp; regression evaluation metrics*, Oct. 2019.

[38] K. Wakefiled, *A guide to machine learning algorithms and their applications*, https://www.sas.com/en_gb/insights/articles/analytics/machine-learning-algorithms.html.

[39] C.-C. Chang and C.-J. Lin, "Libsvm: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, May 2011.

[40] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006, ISBN: 0387310738.

[41] K. J. Cios, R. W. Swiniarski, W. Pedrycz, and L. A. Kurgan, "Unsupervised learning: Association rules," in *Data Mining: A Knowledge Discovery Approach*. Boston, MA: Springer US, 2007, pp. 289–306, ISBN: 978-0-387-36795-8.

[42] MATHWORKS, *Unsupervised learning*, https://www.mathworks.com/discovery/unsupervised-learning.html.

[43] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in *Proc. of 20th Intl. Conf. on VLDB*, 1994, pp. 487–499.

[44] M. J. Zaki, "Scalable algorithms for association mining," *IEEE Trans. on Knowl. and Data Eng.*, vol. 12, no. 3, pp. 372–390, May 2000.

[45] D. Batra, *Lecture 1: Introduction*, https://www.cc.gatech.edu/classes/AY2021/cs7643_fall/slides/L1_intro.pptx.

[46] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., Cham: Springer International Publishing, 2014, pp. 818–833, ISBN: 978-3-319-10590-1.

[47] A. Ng, "The rise of end-to-end learning," in *Machine Learning Yearning*, Draft. deeplearning.ai.

[48] J. Walsh, N. O' Mahony, S. Campbell, A. Carvalho, L. Krpalkova, G. Velasco-Hernandez, S. Harapanahalli, and D. Riordan, "Deep learning vs. traditional computer vision," Apr. 2019, ISBN: 978-981-13-6209-5.

[49] G. Hinton, *Lecture 5: Distributed representations*, http://www.cs.toronto.edu/~bonner/courses/2014s/csc321/lectures/lec5.pdf.

[50] S. Das, B. L. Matthews, and R. Lawrence, "Fleet level anomaly detection of aviation safety data," in *2011 IEEE Conference on Prognostics and Health Management*, IEEE, 2011, pp. 1–10.

[51] S. Das, B. L. Matthews, A. N. Srivastava, and N. C. Oza, "Multiple kernel learning for heterogeneous anomaly detection: Algorithm and aviation safety case study," in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '10, Washington, DC, USA: Association for Computing Machinery, 2010, pp. 47–56, ISBN: 9781450300551.

[52] S. D. Bay and M. Schwabacher, "Mining distance-based outliers in near linear time with randomization and a simple pruning rule," in *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '03, Washington, D.C.: Association for Computing Machinery, 2003, pp. 29–38, ISBN: 1581137370.

[53] L. Li, S. Das, R. John Hansman, R. Palacios, and A. N. Srivastava, "Analysis of flight data using clustering techniques for detecting abnormal operations," *Journal of Aerospace information systems*, vol. 12, no. 9, pp. 587–598, 2015.

[54] S. P. Ackert, *Basics of aircraft maintenance programs for financiers*, http://aircraftmonitor.com/uploads/1/5/9/9/15993320/basics_of_aircraft_maintenance_programs_for_financiers___v1.pdf, Oct. 2010.

[55] A. Altay, O. Ozkan, and G. Kayakutlu, "Prediction of aircraft failure times using artificial neural networks and genetic algorithms," *Journal of Aircraft*, vol. 51, no. 1, pp. 47–53, 2014. eprint: https://doi.org/10.2514/1.C031793.

[56] P. Tamilselvan and P. Wang, "Failure diagnosis using deep belief learning based health state classification," *Reliability Engineering System Safety*, vol. 115, pp. 124–135, 2013.

[57] G. Nicchiotti, "Data-driven prediction of unscheduled maintenance replacements in a fleet of commercial aircrafts," 2018.

[58] S. Zhang and M. Turk, *Eigenfaces*, http://www.scholarpedia.org/article/Eigenfaces, 2008.

[59] H. Lee, H. J. Lim, P. Parker, and A. Chattopadhyay, "Precursor detection of aircraft loss of control in-flight (loc-i) and prediction of future trajectory," in *AIAA AVIATION 2020 FORUM*. eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2020-2879.

[60] R. Deshmukh and I. Hwang, "Anomaly detection using temporal logic based learning for terminal airspace operations," in *AIAA Scitech 2019 Forum*. eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2019-0682.

[61] A. Nielsen, *Practical Time Series Analysis: Prediction with Statistics and Machine Learning*. O'Reilly Media, 2019, ISBN: 9781492041627.

[62] X. Sun, Z. Yang, C. Zhang, G. Peng, and K.-V. Ling, *Conditional gaussian distribution learning for open set recognition*, 2021. arXiv: 2003.08823 [cs.LG].

[63] C. Geng, S.-J. Huang, and S. Chen, "Recent advances in open set recognition: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2020.

[64] M. Verleysen and D. François, "The curse of dimensionality in data mining and time series prediction," in *International work-conference on artificial neural networks*, Springer, 2005, pp. 758–770.

[65] V. Soni and R. Joshi, "A novel dimension reduction technique based on correlation coefficient," *International Journal Of Scientific And Technology Research*, vol. 1, pp. 122–124, 2012.

[66] *Introduction to tensors : Tensorflow core*, https://www.tensorflow.org/guide/tensor.

[67] M. Canizo, I. Triguero, A. Conde, and E. Onieva, "Multi-head cnn–rnn for multi-time series anomaly detection: An industrial case study," *Neurocomputing*, vol. 363, pp. 246–260, 2019.

[68] A. Gozzoli, *Practical guide to hyperparameters optimization for deep learning models*, Jul. 2020.

[69] K. Yoon, "Systems selection by multiple attribute decision making.," 1981.

[70] S. Nanjundan, S. Sankaran, C. Arjun, and G. P. Anand, "Identifying the number of clusters for k-means: A hypersphere density based approach," *arXiv preprint arXiv:1912.00643*, 2019.

[71] C. Sønderby, T. Raiko, L. Maaløe, S. K. Sønderby, and O. Winther, "How to train deep variational autoencoders and probabilistic ladder networks," *ArXiv*, vol. abs/1602.02282, 2016.

[72] S. Lundberg and S.-I. Lee, *A unified approach to interpreting model predictions*, 2017. arXiv: 1705.07874 [cs.AI].

[73] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation functions: Comparison of trends in practice and research for deep learning," *arXiv preprint arXiv:1811.03378*, 2018.

[74] L. Song, *Clustering lecture*, https://www.cc.gatech.edu/~lsong/teaching/CSE6740fall19.html, CSE/ISYE 6740 Class Lecture, Sep. 2019.

[75] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[76] M. Mijwil, "Artificial neural networks advantages and disadvantages," Jan. 2018.

[77] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *arXiv preprint arXiv:1511.08458*, 2015.

[78] M. Memarzadeh, B. Matthews, and I. Avrekh, "Unsupervised anomaly detection in flight data using convolutional variational auto-encoder," *Aerospace*, vol. 7, p. 115, Aug. 2020.